Contributions à la recherche et à l'analyse de modèles 3D

Jean-Philippe Vandeborre

▶ **To cite this version:**

Jean-Philippe Vandeborre. Contributions à la recherche et à l'analyse de modèles 3D. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université des Sciences et Technologie de Lille - Lille I, 2012. tel-00834372

HAL Id: tel-00834372
https://theses.hal.science/tel-00834372

Submitted on 14 Jun 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numéro d'ordre: 40813

Mémoire pour obtenir
l'Habilitation à Diriger des Recherches

en Sciences Mathématiques
spécialité Informatique

par

Jean-Philippe Vandeborre

Contributions à la recherche et à l'analyse de modèles 3D

Soutenu le 15 juin 2012 devant le jury composé de :

M.	Atilla Baskurt	Professeur, INSA Lyon	Rapporteur
M.	Alberto Del Bimbo	Professeur, Università di Firenze, Italia	Rapporteur
Mme	Michela Spagnuolo	Senior Researcher, IMATI-CNR Genova, Italia	Rapporteur
M.	Bruno Lévy	Directeur de Recherche, INRIA Nancy	Examinateur
M.	Christian Roux	Professeur, Télécom Bretagne	Examinateur
Mme	Sophie Tison	Professeur, Université Lille 1	Examinateur
M.	Mohamed Daoudi	Professeur, Télécom Lille 1	Directeur

Université Lille 1 – Sciences et Technologies
LIFL – UMR 8022 CNRS/Lille1

A ma femme, pour sa patience et son amour.
A mes filles, pour leur joie de vivre et leurs sourires.

Remerciements

Je voudrais tout d'abord exprimer mes plus profonds remerciements aux membres du jury. Merci à Atilla Baskurt, Alberto Del Bimbo et Michela Spagnuolo (*grazie mille !*) pour avoir accepté d'être rapporteurs de mon mémoire, malgré un emploi du temps et des responsabilités qui ne les rendent pas toujours faciles à joindre. Leurs avis sur mon travail prouvent que ces dix dernières années de recherche sont reconnues. Merci également à Bruno Lévy, Christian Roux et Sophie Tison d'avoir accepté de faire partie du jury et ainsi témoigner l'intérêt qu'ils portent à mon travail.

Je remercie également Mohamed Daoudi, responsable de l'équipe MIIRE, et collègue avant tout. Les nombreuses discussions que nous avons eues m'ont toujours donné matière à réfléchir et à affiner mes axes de recherche, sans pour autant les diriger. C'est une chance de travailler avec lui.

Une facette importante du métier d'enseignant-chercheur est la formation des jeunes chercheurs. Les guider afin qu'ils acquièrent expérience et autonomie est enrichissant de tous points de vue, aussi bien scientifiquement qu'humainement. Je tiens donc à remercier les doctorants sans qui ce mémoire n'aurait sans doute pas été possible : Tarik Filali Ansary, premier doctorant au caractère bien trempé ; Julien Tierny, impressionnant d'esprit, d'humanité et de modestie ; Halim Benhabiles et Hedi Tabia, acteurs importants de ces trois dernières années de recherche. Merci pour la chance qui m'a été donnée de travailler avec eux. Je leur souhaite le meilleur.

Les projets de recherche auxquels j'ai participé m'ont donné l'occasion de collaborer avec des chercheurs d'autres laboratoires. Sans oublier les autres, je tiens surtout à remercier Guillaume Lavoué du LIRIS avec qui Halim et moi avons travaillé presque quotidiennement pendant les trois années du projet MADRAS.

Ce travail ne serait pas ce qu'il est sans l'excellente ambiance qui règne à Télécom Lille 1 et à l'investissement de chacun dans l'école. Je peux donc

affirmer que tous mes collègues, enseignants-chercheurs, administratifs, techniciens, etc. ont peu ou prou contribué à ce mémoire. Parmi eux, une personne se détache particulièrement : mon collègue de bureau, Jacques Landru, avec qui je partage certainement un peu plus qu'un espace de travail.

Enfin, mes derniers remerciements, sans doute les plus importants, vont vers mes deux petites filles qui me remplissent de fierté tous les jours, et ma chère et tendre épouse qui compte plus que tout au monde.

50/50, ça me paraît raisonnable...

Table des matières

Remerciements **v**

Table des matières **vii**

Liste des figures **x**

Introduction **1**

1 Bilan d'activité **5**
- 1.1 Curriculum Vitæ . . . 5
- 1.2 Activités d'enseignement . . . 6
  - 1.2.1 Principaux enseignements actuels . . . 7
  - 1.2.2 Activités administratives . . . 10
  - 1.2.3 Enseignements passés . . . 11
  - 1.2.4 Enseignements dans d'autres formations . . . 11
- 1.3 Activités de recherche . . . 12
  - 1.3.1 Résumé de ma thèse de doctorat . . . 12
  - 1.3.2 Projets de recherche . . . 13
  - 1.3.3 Encadrements . . . 22
  - 1.3.4 Participation à la communauté scientifique . . . 25
- 1.4 Publications . . . 26

2 Indexation de modèles 3D **35**
- 2.1 Notre approche par vues caractéristiques . . . 36
  - 2.1.1 Présentation rapide de la méthode . . . 37
  - 2.1.2 Descripteur de vues . . . 38
  - 2.1.3 Sélection des vues caractéristiques . . . 42
  - 2.1.4 Indexation probabiliste . . . 46
- 2.2 Résultats et positionnement de notre approche . . . 47
  - 2.2.1 Collection de données . . . 48
  - 2.2.2 Critères d'évaluation . . . 50
  - 2.2.3 Expériences et résultats . . . 50
  - 2.2.4 Quelques résultats utilisant des photographies . . . 55

2.3 Moteur de recherche 3D sur le Web 57
2.4 Conclusion du chapitre 59

3 Analyse structurelle de maillages 3D **61**
3.1 Représentations de formes 3D 62
3.1.1 Représentations volumiques 62
3.1.2 Représentations surfaciques 63
3.1.3 Un peu plus sur les maillages de polygones... 64
3.2 Squelette topologique avancé 65
3.2.1 Résumé de la méthode 66
3.2.2 Graphe de Reeb 66
3.2.3 Quelle fonction d'application ? 68
3.2.4 Sommets caractéristiques 69
3.2.5 Notre fonction d'application 70
3.2.6 Lignes de niveau discrètes et contours discrets 71
3.2.7 Construction du graphe de Reeb de haut niveau 73
3.2.8 Analyse géométrique des contours discrets 74
3.2.9 Obtention d'un squelette topologique à partir d'un graphe de Reeb 76
3.2.10 Complexité et temps de calcul 77
3.3 Trois applications 77
3.3.1 Déformation de maillages 3D 78
3.3.2 Indexation partielle de modèles 3D 78
3.3.3 Segmentation de maillages 3D 82
3.4 Conclusion du chapitre 85

4 Segmentation de maillages 3D **87**
4.1 Segmentation d'un maillage 3D 89
4.2 Pourquoi et comment évaluer des segmentations 3D ? . 89
4.3 Segmentations vérités-terrains 91
4.3.1 Constitution de la collection vérité-terrain 92
4.3.2 Protocole de segmentation manuelle 94
4.3.3 Comparaison avec la collection de l'Université de Princeton 94
4.4 Métriques de comparaison de segmentations 95
4.4.1 Propriétés d'une métrique de comparaison fiable 96
4.4.2 Quelques métriques de comparaison de segmentations . 97
4.4.3 Une nouvelle métrique de comparaison de segmentations 99
4.4.4 Propriétés de la métrique 3D-PRI / 3D-NPRI 101
4.5 Évaluation d'algorithmes de segmentation 104

4.5.1 Les méthodes de segmentations évaluées 104
4.5.2 Résultats 104
4.5.3 Notre benchmark en ligne 106
4.6 Segmentation par apprentissage 108
4.6.1 Notre algorithme de segmentation par apprentissage . . 108
4.6.2 Expériences et résultats 112
4.7 Conclusion du chapitre 118

Conclusion et perspectives **121**
5.8 Résumé et principales contributions 121
5.9 Perspectives 122
5.9.1 Indexation invariante à la pose et par partie 122
5.9.2 Modèles 3D dynamiques (3D+t) 126
5.9.3 Traitement de maillages *bruts* de scanners 3D 127

Bibliographie **131**

Annexes – publications choisies **143**
A. A Bayesian 3-D Search Engine Using Adaptive Views Clustering 144
B. Enhancing 3D mesh topological skeletons with discrete contour constrictions 156
C. Partial 3D Shape Retrieval by Reeb Pattern Unfolding . 175
D. A comparative study of existing metrics for 3D-mesh segmentation evaluation 191
E. Learning boundary edges for 3d-mesh segmentation. . . . 208
E. A new 3D-matching method of non-rigid and partially similar models using curve analysis 222

Liste des figures

1 Evolution de mes activités de recherche depuis ma thèse de doctorat. 3

1.1 Organisation du projet SEMANTIC-3D en sous-projets. . . . 16
1.2 Architecture globale du système SEMANTIC-3D. 17
1.3 Organisation du projet MADRAS en sous-projets ou work-packages. 20
1.4 Vue d'ensemble des différentes tâches du projet et des liens entre les sous-projets. 21

2.1 L'indexation et la recherche de modèles 3D par vues 2D. . . 37
2.2 Quelques pièces mécaniques de la base fournie par Renault dans le cadre du projet SEMANTIC-3D. 40
2.3 Différentes versions d'une porte de Laguna II de la base Renault. 41
2.4 Sélection des vues caractéristiques d'un modèle 3D. 42
2.5 Un modèle 3D *voiture* et l'évolution du score BIC en fonction du nombre de vues caractéristiques. 46
2.6 17 vues caractéristiques d'un modèle 3D *voiture*. 46
2.7 Un cube et trois objets dérivés. 47
2.8 Un objet-requête et les huit premiers résultats par ordre de pertinence. 51
2.9 Courbes rappel-précision pour la méthode AVC avec et sans probabilité sur la collection RNRT SEMANTIC-3D. 52
2.10 Courbes rappel-précision pour quelques méthodes sur le *Princeton Shape Benchmark*. 54
2.11 Requête avec une seule image et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite). 56
2.12 Requête avec deux images et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite). 56

2.13 Requête avec trois images et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite). 56
2.14 Interface web du moteur de recherche en ligne avant toute opération. 57
2.15 Envoi d'une image-requête par l'utilisateur. 58
2.16 Résultats de la recherche à partir de l'image-requête unique. 58
2.17 Résultats de la recherche à partir d'un modèle 3D, résultat de la requête initiale (troisième modèle 3D dans les résultats de la figure 2.16). 58

3.1 *Bimba Con Nastrino* (AIM@SHAPE Shape Repository) en différentes résolutions de maillages. 65
3.2 Principales étapes de notre méthode (de gauche à droite et de haut en bas). 66
3.3 Evolution des lignes de niveau de la fonction hauteur sur un bitore, ses points caractéristiques, et son graphe de Reeb. 67
3.4 Sommets caractéristiques de quelques maillages 3D. 70
3.5 Évolution des lignes de niveau de f_a et ses points critiques (en rouge et en noir). Les points verts sont les sommets caractéristiques. 71
3.6 Exemples de lignes de niveau discrètes sur le modèle *main* composé de 25 000 points : $\Gamma(v_{600})$ présentant 2 contours, $\Gamma(v_{10\,000})$ présentant 4 contours, et $\Gamma(v_{20\,000})$ présentant 6 contours. 72
3.7 Evolution des lignes de niveaux discrètes et des contours discrets sur un tore selon la fonction de hauteur. Une bifurcation – (a) puis (b) – et une jonction – (b) puis (c) – sont détectées. Le graphe de Reeb se construit au fur et à mesure – (d), (e), (f)). 73
3.8 Exemples de graphes de Reeb de différents maillages 3D. . 74
3.9 Graphe de Reeb classique, approximation des constrictions (en rouge), et graphe de Reeb amélioré d'un modèle 3D *cheval*. 75
3.10 Graphe de Reeb classique, approximation des constrictions (en rouge), et graphe de Reeb amélioré d'un modèle 3D *main*. 75
3.11 Exemples de squelettes topologiques obtenus à partir de graphes de Reeb améliorés. 76

3.12 Exemple de déformation d'un maillage 3D. A gauche, le maillage original et son squelette avec ses points de contrôle. A droite, le squelette avec déformations aux articulations et le maillage déformé correspondant. 78
3.13 Requête et résultats dans l'ordre de pertinence. Les appariements de cartes de Reeb sont indiqués par des couleurs identiques. 80
3.14 Appariements de cartes de Reeb entre une main et un humanoïde. 82
3.15 Etapes de la segmentation d'un maillage 3D. 83
3.16 Segmentations hiérarchiques d'un maillage 3D selon les critères topologique. 84
3.17 Segmentations cohérentes à chaque étape de deux objets similaires. 85

4.1 Les vingt-huit modèles 3D de la collection vérité-terrain. Chaque modèle est coloré en fonction d'une de ses vérités-terrains. 92
4.2 Deux segmentations vérités-terrains pour un modèle de cheval vues par deux volontaires différents. 93
4.3 Exemples d'objets 3D extraits de la collection de l'université de Princeton avec leurs vérités-terrains sous forme de frontières (illustration extraite de (Chen et al. 2009)). 95
4.4 Scores 3D-NPRI classés par ordre croissant sur la collection de Chen et al. (2009). 105
4.5 Scores 3D-NPRI classés par ordre croissant sur notre collection de maillages 3D. 105
4.6 Présentation et téléchargement de la collection vérité-terrain sur le site web. 107
4.7 Visualisation des résultats de comparaison des méthodes de segmentation automatique présentes sur le site web. 107
4.8 Classification par la fonction H d'arêtes sur trois modèles 3D. En haut : arêtes frontières en rouge. En bas : valeurs de la fonction H selon l'échelle de couleurs. 110
4.9 Aperçu de la chaîne d'obtention d'une segmentation finale. 111

4.10 Exemples de segmentations automatiques de trois modèles par différentes méthodes. De la gauche vers la droite : moyenne des vérités-terrains du benchmark Princeton (Chen et al. 2009), notre algorithme par apprentissage, algorithmes de Golovinskiy et Funkhouser (2008), de Golovinskiy et Funkhouser (2008), et de Shapira et al. (2008). . . 114
4.11 Segmentations obtenues par notre algorithme d'apprentissage (global sur la collection de l'université de Princeton) pour plusieurs modèles 3D provenant de diverses sources. . 115
4.12 Trois niveau de finesse de segmentation selon le seuil appliqué à la fonction H. 117
4.13 Robustesse de notre algorithme par apprentissage au changement de pose. 118
5.1 Un modèle 3D d'humanoïde dans différentes poses induisant souvent des changements de topologie (cercles rouges). Ces exemples sont tirés de la base TOSCA. 123
5.2 Objet composite : le centaure est une créature mi-homme mi-cheval. 124
5.3 Création d'un nouveau modèle 3D à partir de plusieurs modèles existant. Les parties échangées portent les mêmes couleurs. 125
5.4 Trois étapes d'un maillage dynamique représentant un cheval au galop. 126
5.5 Quatre maillages partiels parmi un total de quatorze obtenus par un scanner Konica-Minolta VIVID910 et son plateau tournant, d'un jouet *Woody*, personnage de *Toy Story* (®Disney). Chaque maillage a une résolution d'environ 50 000 points et 96 000 faces. 128
5.6 Quatorze scans du personnage *Woody* de la figure précédente assemblés automatiquement grâce au logiciel Konica-Minolta *Polygon Editing Tool* fourni avec le scanner. Le maillage final a une résolution de 162 196 points et 253 470 faces. 129

Introduction

Un dessin vaut mieux qu'un long discours.

Proverbe populaire

La représentation numérique des données est en pleine explosion. Images, sons et vidéos sont stockés sur des supports variés, échangés à travers les réseaux mondiaux, produits par des systèmes d'acquisition de plus en plus performants, et manipulés grâce à des outils informatiques puissants et de plus en plus faciles à utiliser.

Depuis quelques années, parmi ces données dites *mutimédia*, il en est une qui prend de plus en plus d'importance et à laquelle tout le monde – du grand public aux professionnels de différents domaines – est confronté quotidiennement ou presque : la représentation tridimensionnelle des objets, appelée plus communément *modèles 3D*. Quoi de plus naturel et intuitif qu'une représentation virtuelle en 3D du monde, lui-même en trois dimensions, qui nous entoure ?

Les représentations tridimensionnelles sont en effet utilisées dans de nombreux domaines. Pour ne prendre que quelques exemples phares, on peut citer :

- l'imagerie médicale utilise des représentations tridimensionnelles du corps humain, en entier ou par partie, pour apporter une vision plus complète et plus précise au diagnostic, et également pour la simulation, la formation des futurs médecins ou l'aide aux opérations chirurgicales ;
- l'industrie mécanique, et notamment automobile, utilise la modélisation informatique depuis de nombreuses années pour le prototypage et le design de nouvelles pièces, ainsi que pour l'assistance à la fabrication des pièces usinées ;
- le tourisme et le patrimoine culturel offrent aux visiteurs des modélisations informatisées de sites, musées, œuvres, etc. Les visiteurs disposent alors d'un point de vue privilégié et d'une interactivité ac-

crue avec les œuvres et autres pièces archéologiques anciennes par exemple ;
- l'industrie des jeux vidéos offre aux joueurs une immersion presque totale dans l'univers du jeu grâce à la modélisation 3D ;
- le cinéma est un grand utilisateur des technologies de modélisation et de traitement tridimensionnel pour la création des effets spéciaux et même la production d'œuvres complètes (la société Pixar par exemple et ses productions les plus connues comme la série des Toy Story).

Cette explosion de la troisième dimension est due, en grande partie, aux améliorations technologiques continues en informatique : augmentation de la puissance de calcul des processeurs, spécialisation des processeurs graphiques, augmentation des capacités mémoires des ordinateurs, accroissement des performances des réseaux de données (en débit et en fiabilité), amélioration des dispositifs de capture tridimensionnelle et des logiciels de modélisation, diminution des coûts matériels, etc.

Les données tridimensionnelles sont représentées de manières différentes en fonction des besoins : nuages de points, représentations surfaciques ou volumiques, etc. Chaque application et chaque logiciel propose également sa propre représentation. Mais c'est certainement la représentation surfacique – qui ne tient compte que de la surface extérieure de l'objet représenté – qui est la plus répandue. Cette représentation surfacique est généralement décrite sous la forme d'un maillage de polygones, qui est en définitive un échantillonnage discret de la surface continue. Le maillage de polygones est parfaitement adapté à l'affichage de l'objet tridimensionnel par la carte graphique d'un ordinateur.

Les besoins concernant ces données augmentent et, surtout, changent depuis quelques années. S'il est de plus en plus aisé de créer des modèles 3D, qu'en est-il de leurs traitements et analyses après création ? Les besoins sont passés de « Comment créer un modèle 3D ? » à « Comment retrouver un modèle 3D similaire à celui ou ceux en ma possession dans une collection de modèles 3D ? » ainsi que « Comment retrouver la structure d'un modèle 3D maillé sans connaissance *a priori* sur celui-ci ? »

Mes thématiques de recherche ont pour but d'apporter des éléments de réponse à ces deux questions. La figure 1 illustre mes travaux de recherche et leurs évolutions depuis ma thèse de doctorat.

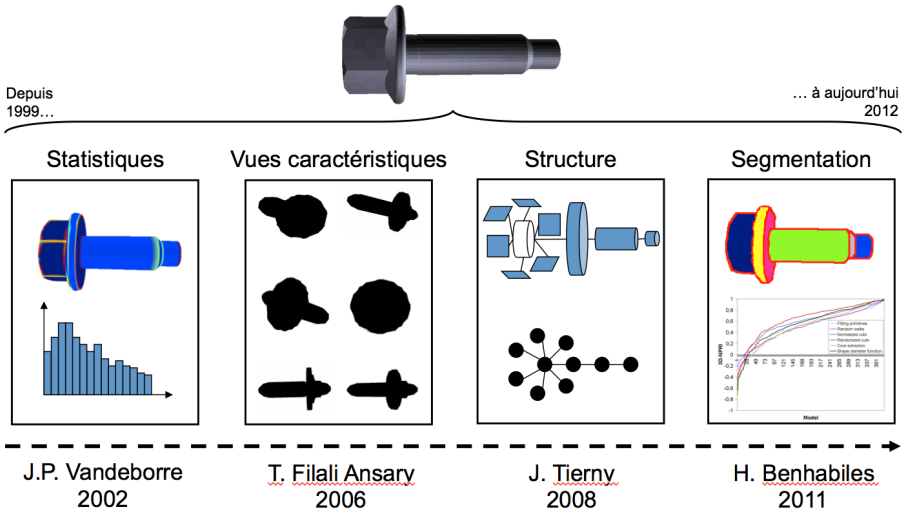


FIGURE 1 – *Evolution de mes activités de recherche depuis ma thèse de doctorat.*

Les travaux que je présente dans ce document s'appuient essentiellement sur l'encadrement des travaux de recherche suivants :

- Thèse de Tarik Filali Ansary (soutenue le 6 octobre 2006), *Recherche de modèles tridimensionnels à partir de vues 2D*. Cette thèse définit un framework complet pour la recherche de modèles 3D à partir d'une ou plusieurs vues 2D, voire d'un modèle 3D complet, par une approche probabiliste. La méthode donne d'excellents résultats (classée 2ème au *Princeton Shape Benchmark* [1]) et a donné le jour à un moteur de recherche en ligne [2], premier moteur au monde en 2006 à proposer une recherche de modèles 3D à partir de photos.
- Thèse de Julien Tierny (soutenue le 2 octobre 2008), *Reeb graph based 3D shape modeling and applications*. Cette thèse propose une modélisation de la structure d'un maillage 3D grâce aux graphes de Reeb et à l'analyse topologique et géométrique. Ce travail théorique a été appliqué à plusieurs domaines pratiques comme la déformation automatique, l'indexation et la segmentation de maillages 3D.
- Thèse d'Halim Benhabiles (soutenue le 18 octobre 2011), *Segmentation de maillages 3D : évaluation automatique des méthodes et nouvelle méthode par apprentissage apprentissage*. La segmentation est une des étapes de pré-traitement de maillage 3D la plus courante. Des méthodes de segmentation commencent à apparaître mais aucune méthode d'évaluation automatique et objective n'existe à l'heure actuelle. Cette thèse a donné naissance à une nouvelle méthode d'éva-

1. http://shape.cs.princeton.edu/benchmark/
2. http://www-rech.telecom-lille1.eu/3dretrieval/

luation des segmentations, ainsi qu'à une nouvelle méthode de segmentation utilisant des connaissances *a priori* sur les modèles 3D par apprentissage

Avec ces considérations, le présent document expose mes activités de recherche et les contributions majeures apportées depuis ma thèse de doctorat et ma nomination en tant que Maître de Conférences de l'Institut Télécom en décembre 2002. Tout au long du document, j'ai essayé de mettre en évidence, outre les résultats obtenus, le cheminement que j'ai suivi pour aller d'une étape vers la suivante dans mes recherches. Ce mémoire se décompose donc de la manière suivante.

Le premier chapitre retrace mes activités d'enseignement et de recherche depuis la fin de ma thèse et mon entrée à l'Institut Télécom.

Le deuxième chapitre apporte des éléments de réponse à la question « Comment retrouver un modèle 3D dans une collection ? ». Cette question est en ligne directe avec ma thèse et en constitue la suite logique.

Le troisième chapitre expose une solution apportée à la question « Comment retrouver la structure d'un modèle 3D maillé ? ». Cette partie a ensuite soulevé deux autres questions traitées dans le chapitre suivant.

Le quatrième chapitre apporte des réponses aux questions posées à la fin du précédent chapitre et présente les travaux sur l'évaluation automatique des méthodes de segmentation de maillages 3D ainsi que sur une nouvelle méthode de segmentation par apprentissage utilisant une vérité-terrain.

La conclusion apporte des perspectives à moyen et long terme pour mes activités de recherche futures en soulevant d'autres questions essentielles dans l'analyse des maillages 3D.

Chaque chapitre résume les réflexions menées, les méthodes développées, et les principaux résultats obtenus dans chacun des domaines. Si le lecteur souhaite connaître plus de détails, des références précises à des publications dont je suis co-auteur sont indiquées aux endroits adéquats. Des copies de ces publications, que je juge essentielles dans mon activité de chercheur, sont par conséquent données en annexes.

Bilan d'activité

1

Je suis actuellement Maître de Conférences de l'Institut Télécom [1] en poste à Télécom Lille 1 [2] où j'effectue mes activité d'enseignement. Je fais partie de l'équipe MIIRE du Laboratoire d'Informatique Fondamentale de Lille (LIFL [3] – UMR 8022 Lille1/CNRS) de l'Université Lille 1 [4], et plus particulièrement de l'équipe MIIRE [5] de ce laboratoire. Ce chapitre dresse un bilan rapide de mes activités depuis la fin de ma thèse de doctorat. Il résume l'ensemble de mes activités d'enseignant-chercheur. La liste exhaustive de mes publications est donnée en fin de chapitre.

1.1 Curriculum Vitæ

Etat civil

Vandeborre Jean-Philippe
Né le 19 mars 1974 à Tourcoing (59 – France)
Nationalité française
Marié, deux enfants

Situation professionnelle

Emploi :	Maître de Conférences de l'Institut Télécom en poste à Télécom Lille 1
Date de recrutement :	2 décembre 2002
Adresse professionnelle :	Télécom Lille 1 – bureau D112S rue G. Marconi – cité scientifique 59655 Villeneuve d'Ascq cedex
Téléphone direct :	+33 3 20 33 55 96
Adresses méls :	jean-philippe.vandeborre@telecom-lille1.eu jean-philippe.vandeborre@lifl.fr

1. http://www.institut-telecom.fr/
2. http://www.telecom-lille1.eu/
3. http://www.lifl.fr/
4. http://www.univ-lille1.fr/
5. http://www-rech.telecom-lille1.eu/miire/

Formation

2002 :	Doctorat en Informatique de l'Université Lille 1. "Modèles 3D : indexation, et habillage par textures extraites de photographies", sous la direction de Mohamed Daoudi et Christophe Chaillou, thèse soutenue le 22 novembre 2002
1997 :	DEA en Informatique de l'Université Lille 1
1995-1996 :	Licence et Maîtrise d'Informatique de l'Université Lille 1
1994 :	DEUG A (Sciences et Structures de la Matière) de l'Université Lille 1
1992 :	Baccalauréat C

Situations antérieures

1999 - 2002 :	Enseignant-chercheur à Télécom Lille 1 (ex-ENIC)
1997 - 1999 :	Doctorant boursier MESR au LIFL

1.2 Activités d'enseignement

Depuis mon arrivée à Télécom Lille 1 (ex-ENIC [6]), en tant qu'enseignant-chercheur du GIE ENIC de décembre 1999 à décembre 2002 puis en tant que Maître de Conférences de l'Institut Télécom depuis décembre 2002, je fais partie du département Informatique et Réseaux.

Télécom Lille 1 est une école d'ingénieurs en Sciences et Technologies de l'Information et de la Communication. Elle délivre un diplôme d'Ingénieur reconnu par la Commission des Titres d'Ingénieurs (CTI). Elle est également membre actif de la Conférence des Grandes Ecoles (CGE).

La formation d'ingénieurs de Télécom Lille 1 s'adresse à un public étudiant très varié par les différents niveaux de recrutement : formation initiale (recrutement post-Baccalauréat par le concours GEIPI pour les bacs S ou concours interne pour les bacs STI, cinq années de formation avec de nombreux stages), formation par apprentissage (recrutement post-Bac+2, DUT réseau par exemple, puis trois années de formation en apprentissage dans une entreprise), formation continue (recrutement de professionnels avec au moins quatre années d'expérience) présentielle (une année de for-

6. A sa création en 1990, l'école s'appelait ENIC (École Nouvelle d'Ingénieurs en Communication) et faisait partie des NFI (Nouvelles Formations d'Ingénieurs). Elle est le fruit de la collaboration de l'Institut Télécom et de l'Université Lille 1, et se trouve sur le campus de cette dernière. Elle a, par la suite, changé de nom en 2001 pour devenir ENIC Télécom Lille 1, puis finalement en décembre 2005, Télécom Lille 1.

mation à temps plein suivie de six mois de projet de fin d'étude) ou distante (format TuTelNet[7], trois ans de formation à temps partiel, en parallèle de l'activité professionnelle) et formation à la carte en fonction des demandes de clients (France Télécom, SNCF, etc.)

Dans la suite de cette section, je décris mes principaux enseignements actuels, mes activés plus administratives au sein de l'école, mes enseignements passés, et enfin mes enseignements dans d'autres formations.

1.2.1 Principaux enseignements actuels

En tant qu'enseignant-chercheur du département Informatique et Réseaux de Télécom Lille 1, j'effectue des enseignement dans de nombreux domaines de l'informatique et des réseaux. En particulier, mes principaux thèmes d'enseignement sont les systèmes d'exploitation Unix, les protocoles de l'Internet, et bien sûr, la 3D en lien avec mes activités de recherche (synthèse d'images, modélisation, etc.)

Coordinations et responsabilités principales

A Télécom Lille 1, la coordination d'une UV ou d'un module d'enseignement est une tâche complète qui comporte la création du cours, des travaux dirigés et pratiques, la conception des supports, la gestion des emplois du temps et la réservation des salles et amphithéâtres, la coordination des différents intervenants, la conception et la correction des examens, la participation aux jurys, etc. Les temps d'enseignements donnés dans ce paragraphe s'entendent en face-à-face pédagogique, c'est-à-dire hors coordination, corrections, etc.

Le tableau ci-dessous résume mes activités dans mes principaux modules. Il indique, pour chaque UV principale, le nombre moyen d'étudiants concerné chaque année (qui varie selon la taille de la promotion), le nombre d'heures que représente cette UV du point de vue de l'étudiant, le nombre d'heures d'interventions (cours/TD/TP) que j'effectue personnellement dans cette UV, et le nombre d'intervenants impliqués et à gérer dans le déroulement global de l'UV.

7. http://www.telecom-lille1.eu/sites/default/files/upload/PDF/Formation_continue.pdf

UV ou module	effectif étudiant	h. vues de l'étudiant	h. d'intervention perso.	intervenants à gérer
TSIO	80	120	40	10
TA2	80	30	12	5
Synthèse 3D	10	30	30	1

Je suis le concepteur et le coordinateur de l'UV *Technologies des Systèmes d'Information Ouverts* (TSIO). TSIO est une UV obligatoire pour les étudiants de quatrième année (Bac+4, deuxième année du cycle ingénieur) visant à donner un socle commun de connaissances indispensables en informatique pour aborder les UV de spécialisation qui suivront dans le cursus. TSIO aborde trois grands thèmes de base de l'informatique pour les systèmes d'information ouverts et interopérables : les systèmes d'exploitation et les réseaux, les bases de données, et les technologies web-java. J'interviens dans le thème *systèmes d'exploitation et réseaux*. Les deux autres thèmes sont pris en charge par des collègues experts des domaines en question. L'UV TSIO représente 120 heures d'enseignement pour l'étudiant. La partie *systèmes d'exploitation et réseaux* représente environ 40 heures d'enseignement que j'effectue intégralement.

Je suis également le concepteur et le coordinateur du module de technologie avancée TA2 *Introduction à Unix et à la programmation système en langage C*. Le module TA2 fait partie des modules de technologie essentiellement axés sur la pratique : quelques heures de cours puis des travaux pratiques sous la forme d'un projet complet. Ce module est dispensé en troisième année (Bac+3, première année du cycle ingénieur) et en double, pour chaque demi-promotion. Il représente 30 heures d'enseignement pour l'étudiant, le double pour l'enseignant. Je fais les cours (4,5 heures $\times$ 2) et les évaluations de projets en fin de chaque semaine (1,5 heures $\times$ 2) du module. Les TP et projets sont réalisés principalement par des vacataires que je coordonne.

Enfin, j'ai créé et je coordonne la partie *synthèse d'images 3D* de l'UV de spécialisation *3D Digital Entertainment Technologies* (3DETech). Cette UV présente les méthodes et techniques permettant d'interagir avec les environnements numériques et virtuels, et tout ce qui tourne autour des loisirs numériques en 3D. Une grande partie de cette UV est consacrée aux techniques de vision 3D et de synthèse d'images qui seront au cœur des services et des usages de demain. La partie *synthèse d'images 3D* représente environ 30 heures d'enseignement que j'effectue intégralement.

Principales participations dans d'autres modules

En plus des enseignements dans mes domaines de prédilection, je participe aussi aux travaux dirigés et pratiques d'autres modules coordonnés par des collègues :

- UV de spécialisation *Architectures Réseaux* (ARES) – 5e année ;
- module commun A42 *Réseaux : modèles, architectures et protocoles* – 3e année ;
- module commun A31 *Structures de données en C* – 3e année ;
- module commun A23 *Architecture des ordinateurs* – 3e année.

Je participe également aux enseignements à distance (format TuTelNet) de l'UV de spécialisation ARES : regroupement de cours et travaux pratiques.

Le tableau suivant résume mes activités dans ces modules.

Module	**année**	**effectif étudiant**	**nb. étudiants par groupe**	**nb. heures effectuées**
ARES	5e	20 à 25	—	6 h. TP
A42	3e	80 à 90	25 à 30	12 h. TD/TP
A31	3e	80 à 90	25 à 30	12 h. TD/TP
A23	3e	80 à 90	25 à 30	12 h. TD/TP
ARES-TTN	TuTelNet	25 à 30	—	4 h. cours + 6 h. TP

Suivi d'étudiants et projets technologiques

Télécom Lille 1 proposant une formation par apprentissage aux étudiants titulaires d'un Bac+2 dans le domaine des TIC (par exemple le DUT GEII), je suis tuteur école de plusieurs apprentis. L'apprentissage est d'une durée de trois ans alternant périodes en entreprise et cours à distance. Le tuteur école veille à l'évolution des missions de l'apprenti tout au long de ces trois années, rend visite à l'apprenti et à son maître d'apprentissage au moins deux fois par an, évalue les rapports annuels d'activité et les soutenances. Je suis plusieurs apprentis tous les ans.

Pendant leur cursus, pour compléter les cours au sens traditionnel du terme (cours en amphithéâtre, travaux dirigés et pratiques) et les stages en entreprise, les étudiants suivent également des enseignements sous forme de projets, appelés *Projets Scientifiques et Technologiques*. Tous les ans, je propose et suis plusieurs projets de ce genre. Les sujets sont toujours

orientés vers mes activités de recherche. Cela permet à l'étudiant de se sentir impliqué dans un projet pédagogique avec une utilité directe pour l'enseignant-chercheur et non un simple devoir pédagogique sans autre utilité que de donner une note.

Enfin, je suis également plusieurs étudiants de filière initiale en projet de fin d'études (stage finale en cinquième année) tous les ans, en fonction des besoins.

1.2.2 Activités administratives

Les principales responsabilités administratives des enseignants-chercheurs à Télécom Lille 1 sont les coordination de modules et UV d'enseignement. Comme je l'ai déjà souligné, il s'agit là d'un travail important puisqu'il ne se résume pas à donner les cours, mais nécessite une prise en charge complète de l'enseignement, de la création de l'emploi du temps, jusqu'aux corrections de copies et à la participations aux jurys.

En plus de ces coordinations, je participe chaque années aux entretiens de sélection des nouveaux étudiants, toutes formations confondues. Je participe également aux réunions de validation de stages et de projets d'apprentissage.

J'ai été représentant des personnels au Conseil d'Orientation de TELECOM Lille 1 (comité d'évaluation des actions passées et de propositions d'évolutions de l'école) de 2002 à 2007 inclus.

Depuis 2010, je suis représentant de TELECOM Lille 1 au Pôle Images Nord-Pas de Calais [8], comité des formations. Je participe notamment aux discussions relatives à l'évolutions des formations autour de l'image dans le Nord-Pas de Calais et à leurs collaborations futures.

Enfin, en tant qu'utilisateur et fervent partisan des logiciels libres [9], je contribue à la diffusion du système d'exploitation Linux et d'autres logiciels libres dans mes enseignements et en apportant mon expertise au Support Informatique et Réseaux de Télécom Lille 1. De plus, j'ai eu le plaisir d'organiser deux conférences sur ce thème à Télécom Lille 1 à destination du public dans le cadre des conférences RMIT (Rencontres autour des Métiers de l'Informatique et des Télécommunications) : « Logiciels libres et formats ouverts en entreprise » en juin 2007 [10] et « Economie et

8. http://www.pole-images-nordpasdecalais.com/
9. http://fr.wikipedia.org/wiki/Logiciel_libre
10. http://www.telecom-lille1.eu/people/vandeborre/confLL1.html

marché du logiciel libre en France » en mars 2010 [11]. Chacune de ces deux manifestation a attiré une centaine de personnes, du grand public aux professionnels du domaine.

1.2.3 Enseignements passés

Comme je l'ai indiqué en introduction de cette partie concernant mes enseignements, depuis que je suis à Télécom Lille 1, j'ai eu l'occasion d'enseigner à tous les niveaux de formation proposés à l'école.

Par le passé, j'ai donc été en charge du module d'initiation à l'algorithmique et à la programmation en première année et du module sur les réseaux en troisième année. Ceci aussi bien en formation initiale présentielle qu'en formation à distance TutelNet.

J'ai surtout été le concepteur et le coordinateur de l'option de spécialisation scientifique et technologique (*Infrastructures Systèmes et Intégration des Services – ISIS*). Cette option concernait l'ensemble des services (rendus par l') informatique en entreprise à l'aide de technologies ouvertes et interopérables. Depuis la rentrée 2010 et la refonte du cursus final de spécialisation, l'option ISIS, comme toutes les options de spécialisation, n'existe plus. Son contenu a servi de base à l'élaboration de la nouvelle UV obligatoire TSIO (voir page 8).

Enfin, j'ai participé à diverses formations "sur mesure" comme le DU de l'Université Lille 1 géré par Télécom Lille 1, *Expert IP* pour France Télécom. Il s'agissait d'enseignement à distance au format TuTelNet sur les réseaux Internet.

1.2.4 Enseignements dans d'autres formations

Pour terminer, je donne également des enseignements en dehors de Télécom Lille 1.

Depuis 2006, je suis responsable du module *Introduction to Computer Networks* du Master Global E-Business [12] de l'Université Lille 1. Ce cours vise à donner les bases des réseaux d'ordinateurs en une trentaine d'heures à des étudiants venant de formations non-informatiques. Les étudiants viennent de nombreux pays (notamment la Chine et les pays d'Europe de l'Est), les cours se font donc en anglais.

11. http://www.telecom-lille1.eu/people/vandeborre/confLL2.html
12. http://global-ebusiness.univ-lille1.fr/

J'ai également donné des cours d'indexation de modèles 3D en Master Recherche en Informatique de l'Université Lille 1, en lien direct avec mes activités de recherche.

1.3 Activités de recherche

Mon domaine de recherche concerne les modèles tridimensionnels, appelés plus communément modèles 3D, et plus particulièrement l'indexation l'analyse et la segmentation des modèles 3D. Le chalenge qui m'a toujours motivé, et ceci depuis mon DEA, est de retrouver de l'information et une certaine structure dans les modèles 3D maillés qui sont, par définition, constitués d'une "soupe" de polygones sans autre information que la position des points du maillage dans l'espace et leurs relations pour former des triangles ou d'autres polygones.

Dans cette section, je résume en quelques phrases ma thèse de doctorat. Puis je décris mes activités depuis la thèse à travers les projets auxquels j'ai participé et les encadrements d'étudiants en Master Recherche et en thèse. Je termine par mes participations aux comités de conférences et à la relecture d'articles pour des revues et conférences.

1.3.1 Résumé de ma thèse de doctorat

Dans ma thèse, j'apporte des éléments de réponse à deux questions auxquelles doit souvent faire face un concepteur de modèles tridimensionnels : « Comment retrouver un ou plusieurs modèles 3D dans une collection en présentant une requête sous la forme d'un exemple ? » et « Comment habiller les modèles de façon à ce que leurs rendus soient le plus réaliste possible ? »

J'ai tout d'abord proposé une méthode d'indexation de modèles 3D par des descripteurs de forme, et, d'autre part, une approche de l'habillage d'un modèle 3D connu par des textures extraites de photographies le représentant mais dont les conditions de prise de vue sont inconnues.

La méthode d'indexation proposée s'attachait à décrire les modèles 3D maillés par trois descripteurs de forme. Ces trois descripteurs sont des distributions de caractéristiques de formes locales ou globales. Le premier descripteur correspond à la distribution des courbures calculées en tout point du maillage. Le second est une distribution stochastique des dis-

tances euclidiennes entre les points du modèle. Le troisième est une distribution des volumes élémentaires tétraédriques composant le modèle. Des méthodes de combinaison des résultats de chaque descripteur ont été présentées dans le but de ne conserver que les meilleurs résultats de chacun d'eux.

Le réalisme des images de synthèse étant le plus souvent obtenu par placage de textures, ces dernières doivent provenir de sources elles-mêmes réalistes comme des photographies d'objets réels. La seconde contribution de la thèse réside dans une méthode d'extraction de textures à partir de photographies. Les modèles 3D étant largement disponibles (bases de données, Internet etc.), il est possible d'obtenir un modèle 3D d'un objet représenté sur une photographie mais il est souvent impossible de connaître les conditions de prise de vue de la photographie. J'ai donc proposé une méthode d'extraction des textures visibles sur une photographie en recalant le modèle sur la photographie, et en retrouvant les conditions de prise de vue de celle-ci.

1.3.2 Projets de recherche

Depuis la thèse, je me suis concentré sur la partie indexation et analyse de modèles 3D puisque c'est ce domaine qui présente le chalenge auquel je m'intéresse : **trouver de l'information dans les modèles 3D maillés**.

Tout d'abord, je résume en quelques mots mes activités de recherche avant de donner plus de détails dans les paragraphes qui suivent.

Pour répondre au défi de trouver de l'information dans les modèles 3D maillés, je me suis tout d'abord intéressé à l'indexation de modèles 3D à partir de vues 2D (DEA puis thèse de Tarik Filali Ansary, financée par le projet RNRT SEMANTIC-3D, soutenue en octobre 2006). Puis est venue la question de l'information structurelle qui conduit à créer un modèle 3D maillé mais qui, bien évidemment, disparaît une fois celui-ci créé. L'analyse structurelle et topologique a donc été mon second sujet d'intérêt, d'autant plus qu'il me trottait dans la tête depuis plusieurs années. Le Master Recherche et la thèse de Julien Tierny, financée par une bourse du Ministère de l'Enseignement Supérieur et de la Recherche français, et soutenue en octobre 2008, m'ont permis de diriger mes premiers travaux sur le sujet. La segmentation de maillages 3D étant très souvent une étape de pré-traitement avant l'indexation et une des applications directes de l'analyse structurelle, je me suis intéressé à ce domaine. Cependant,

les publications sur le sujet manquent toutes d'un élément fédérateur : un benchmark pour comparer objectivement les méthodes entre elles. Le Master Recherche puis la thèse – financée par le projet ANR MADRAS, soutenue le 18 octobre 2011 – d'Halim Benhabiles portaient justement sur l'élaboration d'une collection de données avec vérités-terrains et également sur des critères objectifs de comparaison des méthodes. Enfin, pour continuer dans l'indexation de modèles 3D, je travaille sur l'indexation partielle, c'est-à-dire la possibilité d'interroger une base de modèles 3D en ne présentant qu'un morceau de modèle 3D en requête. Ce type d'indexation n'est encore que très peu exploré dans la littérature. La thèse d'Hedi Tabia, bourse MESR, soutenue le 27 septembre 2011, et la thèse en cours de Rachid El Khoury (bourse Futur et Ruptures de l'Institut Télécom) portent sur ce sujet.

Projet RNRT SEMANTIC-3D (novembre 2002 à janvier 2006)

- **implication :** participant ;
- **encadrement :** thèse de Tarik Filali Ansary (co-encadrement avec Mohamed Daoudi) soutenue le 6 octobre 2006 ;
- **livrables :** co-rédacteurs de trois livrables ;
- **publications :** co-auteur d'un chapitre de livre aux éditions Willey and Sons, co-auteur de deux articles dans des journaux internationaux, de cinq articles dans des conférences internationales, et de quatre articles dans des conférences nationales.
- **autres résultats :** développement d'un des premiers moteur de recherche de modèles 3D à partir de vues 2D sur le web (voir chapitre 2, section 2.3, page 57 de ce document).

Dès ma nomination en tant que Maître de Conférences, j'ai participé très activement au projet SEMANTIC-3D [13]. Il s'agissait d'un projet exploratoire dans le cadre du Réseau National de Recherche en Télécommunications (RNRT) labellisé en 2002. Il a débuté en novembre 2002 pour se terminer en janvier 2006. Ce projet exploratoire regroupait six partenaires, quatre académiques et deux industriels : le LIRIS [14] (Laboratoire d'InfoRmatique en Image et Systèmes d'information de l'Université Lyon 1), Télécom Lille 1 (ex-ENIC), EURECOM [15] (école de l'Institut Télécom à Sophia-Antipolis), le projet Artemis [16] de Télécom & Management Sud-

13. http://liris.cnrs.fr/semantic-3d/
14. http://liris.cnrs.fr/
15. http://www.eurecom.fr/
16. http://www-artemis.it-sudparis.eu/

Paris (ex-INT à Evry), Renault (constructeur automobile) et TGS (PME spécialisée dans la visualisation 3D, rachetée par Mercury Computer System Inc. au cours du projet).

Le projet SEMANTIC-3D – *Service d'Échange et de MANipulation (Tatouage, Indexation et Compression) des objets 3D* – avait pour objet l'exploration de techniques et d'outils préalables à la réalisation de nouveaux services opérationnels pour l'exploitation de contenu 3D à travers le Web et les objets communicants. En particulier, ce projet visait à permettre un accès facile et rapide aux objets 3D d'une base de données avec des méthodes d'indexation spécifiques pour retrouver des objets similaires ; améliorer l'interactivité des utilisateurs de ces services avec des objets 3D en proposant des fonctionnalités nouvelles (expression d'une requête 3D, décomposition de l'objet visualisé) ; accélérer la transmission et la visualisation des objets 3D en développant des méthodes adaptées de compression/décompression ; assurer la protection des données et/ou intégrer de l'information supplémentaire en tatouant les objets 3D ; envisager la variété des ressources disponibles, incluant les terminaux mobiles ; prototyper un système opérationnel intégrant ces techniques, au niveau serveur, format d'échange et client, prenant en compte la dimension utilisateur et l'adaptation des données sources.

Avec ces objectifs, le projet a été découpé en quatre sous-projets : SP1 – compression, transmission, visualisation ; SP2 – indexation, identification ; SP3 – tatouage ; SP4 – spécifications, intégration, validations, démonstration. La figure 1.1 montre ce découpage du projet en sous-projets. Notre équipe était responsable du sous-projet SP2.

Le projet SEMANTIC-3D a permis le financement de la thèse de doctorat de Tarik Filali Ansary soutenue en octobre 2006. Nous avons proposé une nouvelle méthode d'indexation de modèles 3D à partir de vues caractéristiques et d'une approche probabiliste bayesienne. Ce travail fait l'objet du chapitre 2 du présent document. Le projet a également été le début d'une collaboration durable entre le LIRIS et notre équipe puisqu'il a facilité notre rapprochement et la rédaction du projet ANR MADRAS (cf. page suivante).

Le résultat final du projet a également été la réalisation d'un prototype de système complet. Ce système a été élaboré en suivant des scénarios d'usages établis par Renault. Par exemple, un réparateur en concession ou en dépannage interroge le central des méthodes de réparation par des mots-clefs, des images, voire des objets 3D s'il en a en sa possession. La

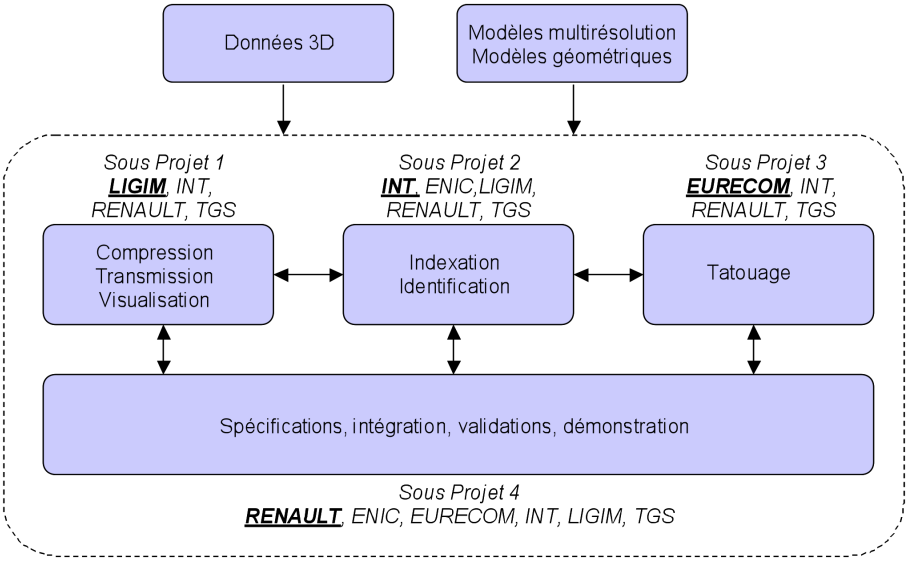


FIGURE 1.1 – *Organisation du projet SEMANTIC-3D en sous-projets.*

méthode de réparation lui est alors envoyée sous forme de scènes animées, permettant de faire la réparation de manière optimale. L'architecture globale du projet (Baskurt et al. 2004) est illustrée par la figure 1.2.

Enfin, outre les nombreuses publications scientifiques, l'ultime conclusion du projet SEMANTIC-3D a été la publication d'un livre sur la compression, l'indexation et le tatouage d'objets 3D aux éditions Wiley & Sons (Dugelay et al. 2008) en avril 2008.

Bourse MESR pour l'analyse structurelle de maillages 3D

- **implication :** rédacteur du sujet de thèse, demande auprès du MESR ;
- **encadrement :** master recherche (encadrement seul) puis thèse de Julien Tierny (co-encadrement avec Mohamed Daoudi) soutenue le 2 octobre 2008 ;
- **publications :** co-auteur de deux articles dans des journaux internationaux, de cinq articles dans des conférences internationales, et de trois articles dans des conférences nationales.

Suite au projet SEMANTIC-3D et aux résultats obtenus en indexation 3D, la question de la structure cachée dans les maillages était toujours présente et de plus en plus prégnante. En 2005, j'ai obtenu un financement du Ministère de l'Enseignement Supérieur et la Recherche français pour Julien Tierny. La thèse a été soutenue le 2 octobre 2008.

La liberté de recherches offerte par une bourse ministérielle nous a

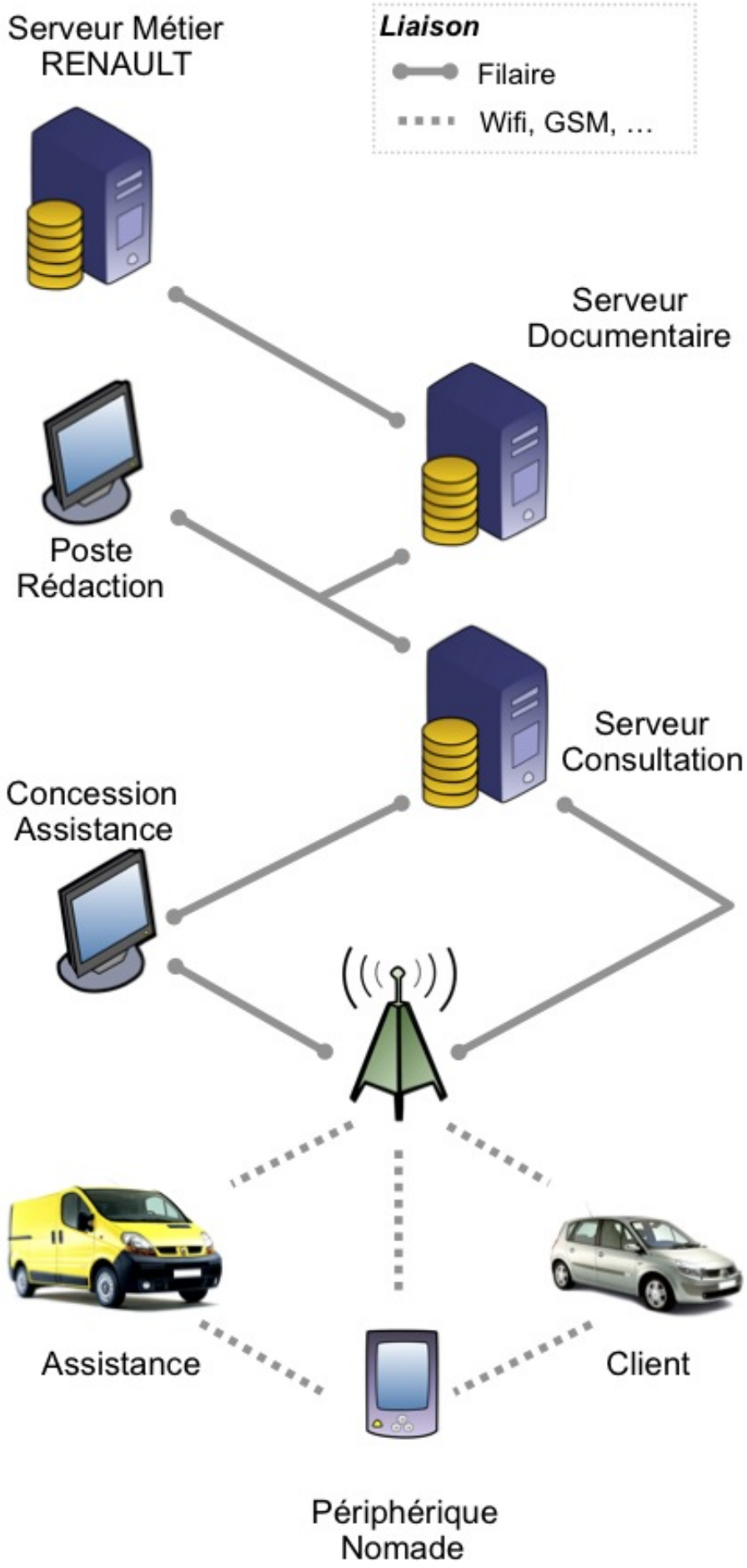


FIGURE 1.2 – *Architecture globale du système SEMANTIC-3D.*

permis d'obtenir des résultats théoriques significatifs avec l'utilisation de la théorie de Morse et les graphes de Reeb ainsi que des applications pratiques qui ont ouvert les portes vers d'autres sujets, notamment la segmentation de maillages. Ces travaux sur l'analyse structurelle de maillages font l'objet du chapitre 3 du présent mémoire.

DELOS Network of Excellence (2006 - 2008)

- **implication :** participant ;
- **livrables :** co-rédacteurs d'un livrable ;
- **publications :** co-auteur de quatre articles dans des conférences et workshops internationaux.

Le Réseau d'Excellence Européen DELOS [17] était financé dans le cadre du programme pour les technologies de la société de l'information (Information Society Technologies program – IST) de la Commission Européenne. Le réseau DELOS coordonnait les recherches de plus de 55 partenaires dans le domaine des bibliothèques numériques du futur. Les activités étaient nombreuses : architecture des nouvelles bibliothèques numériques, accès à l'information, données multimédia complexes, extraction de connaissances et sémantique, interfaces utilisateurs et visualisation, etc.

Nous avons participé plus particulièrement au sous-projet WP3 – Audio/Visual and Non-traditional Objects. Nous y avons poursuivi nos travaux concernant l'indexation ainsi que ceux concernant l'analyse structurelle.

Le Réseau d'Excellence DELOS a commencé le 1er janvier 2004 pour une durée de 48 mois. Nous avons été partenaire de ce réseau de 2006 à 2008. Maintenant, le Réseau d'Excellence en tant que tel n'existe plus et a été transformé en une association avec les mêmes buts de fédération de la recherche européenne dans le domaine des bibliothèques numériques multimédia.

Projet ANR MADRAS (janvier 2008 à septembre 2011)

- **implication :** co-rédacteur de la proposition auprès de l'ANR, responsable local LIFL/Université Lille1 tout au long du projet ;
- **encadrement :** Mater Recherche (encadrement seul) puis thèse d'Halim Benhabiles (co-encadrement avec Mohamed Daoudi et Guillaume Lavoué (INSA Lyon/LIRIS)) soutenue le 18 octobre 2011 ;

17. http://www.delos.info/

- **livrables :** co-rédacteurs de trois livrables, des rapports annuels et du rapport final de projet, co-présentation des évaluations à mi-parcours et finales du projet ;
- **publications :** co-auteur de deux articles dans des journaux internationaux, de deux articles dans des conférences internationales, et de trois articles dans des conférences nationales.
- **autres résultats :** mise en place d'un benchmark en ligne pour l'évaluation automatique des méthodes de segmentation de mailalges 3D (voir chapitre 4, section 4.5.3, page 106 de ce document).

Quand j'ai commencé à m'intéresser à la question de l'analyse structurelle des maillages 3D, j'avais également en tête la question de la segmentation de ces maillages. En effet, une étape préliminaire à de nombreux autres traitement du maillages, comme l'indexation ou la compression, est la segmentation. La segmentation consiste à subdiviser le maillage en différentes parties aux caractéristiques identiques, soit d'un point de vue strictement géométrique, soit d'un point de vue perceptif et/ou sémantique. De nombreux systèmes ont été, et sont encore actuellement, développés pour la segmentation d'images ou de vidéos. Mais ces méthodes ne sont pas adaptables à la 3D. De plus, on constate dans la littérature que, contrairement à la segmentation d'images 2D, il n'existe ni protocole ni jeu de données standard pour la comparaison et l'évaluation des méthodes de segmentations de maillages 3D.

Le projet MADRAS [18] – *3D Models And Dynamic models Representation And Segmentation* – a été financé par l'Agence Nationale de la Recherche (ANR) dans le cadre de l'appel à projet *Masse de Données et COnnaissances* (MDCO) 2007 pour une durée de trois ans à partir du 1er janvier 2008. Il a ensuite été prorogé jusqu'à la fin du mois de septembre 2011. Il s'agit d'un projet de recherche académique puisqu'il ne concerne aucun partenaire industriel. Les partenaire du projet MADRAS sont le LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information de l'Université Lyon 1), l'INRIA Rhône-Alpes, et Télécom Lille 1 / Université Lille 1. Le projet est porté par le LIRIS. Je suis le responsable côté Télécom Lille 1 / Université Lille 1.

Le projet MADRAS a poursuivi un triple objectif :

1. Créer une collection de maillages 3D et 3D+t – c'est-à-dire de maillages 3D statiques et dynamiques – accompagnée d'une vérité-terrain. Cette vérité-terrain est composée d'une ou plusieurs segmen-

18. http://www-rech.telecom-lille1.eu/madras/

tations pour chaque modèle 3D. Ces segmentations viennent de segmentations "faites-mains" mais aussi de méthodes automatiques ou semi-automatiques.

2. Exploiter le facteur humain, au travers de tests subjectifs, pour améliorer la conception et l'évaluation des algorithmes de segmentation. Les aspects subjectifs et perceptifs sont utilisés pour mettre en place des outils de référence qui permettront un processus automatique de comparaison des méthodes de segmentation existantes ou futures.
3. Mettre en œuvre de nouveaux algorithmes de segmentation aussi bien pour les maillages statiques (3D) que dynamiques (3D+t) exploitant le facteur humain, par des techniques d'apprentissage.

Pour répondre à ces trois objectifs, le projet MADRAS a été découpé en quatre sous-projets ou *workpackages* (sous-projet de gestion du projet lui-même inclus) comme illustré à la figure 1.3.

La figure 1.4 donne une vue d'ensemble des différentes tâches et liens entre sous-projets, qui étaient prévus dans le projet MADRAS.

Le projet MADRAS a donné d'excellents résultats aussi bien au niveau de l'évaluation des méthodes de segmentation que d'une nouvelle méthode de segmentation par apprentissage. Ce travail fait partie de la thèse, financée par le projet, d'Halim Benhabiles et fait l'objet du chapitre 4 de ce document.

Subproject	Leader	Other partners
WP0 - Project organization and management	LIRIS	USTL/LIFL INRIA R.A.
WP1 -3D-models collection with ground-truth segmentations	USTL/LIFL	LIRIS INRIA R.A.
WP2 -Subjective analysis and evaluation	LIRIS	USTL/LIFL INRIA R.A.
WP3 -New segmentation methods and evaluation	INRIA R.A.	LIRIS USTL/LIFL

FIGURE 1.3 – *Organisation du projet MADRAS en sous-projets ou workpackages.*

Bourse *Futur et Rupture* pour l'indexation par partie

Comme je l'ai indiqué en introduction de la présente section, les recherches en indexation de modèles 3D s'orientent de plus en plus vers

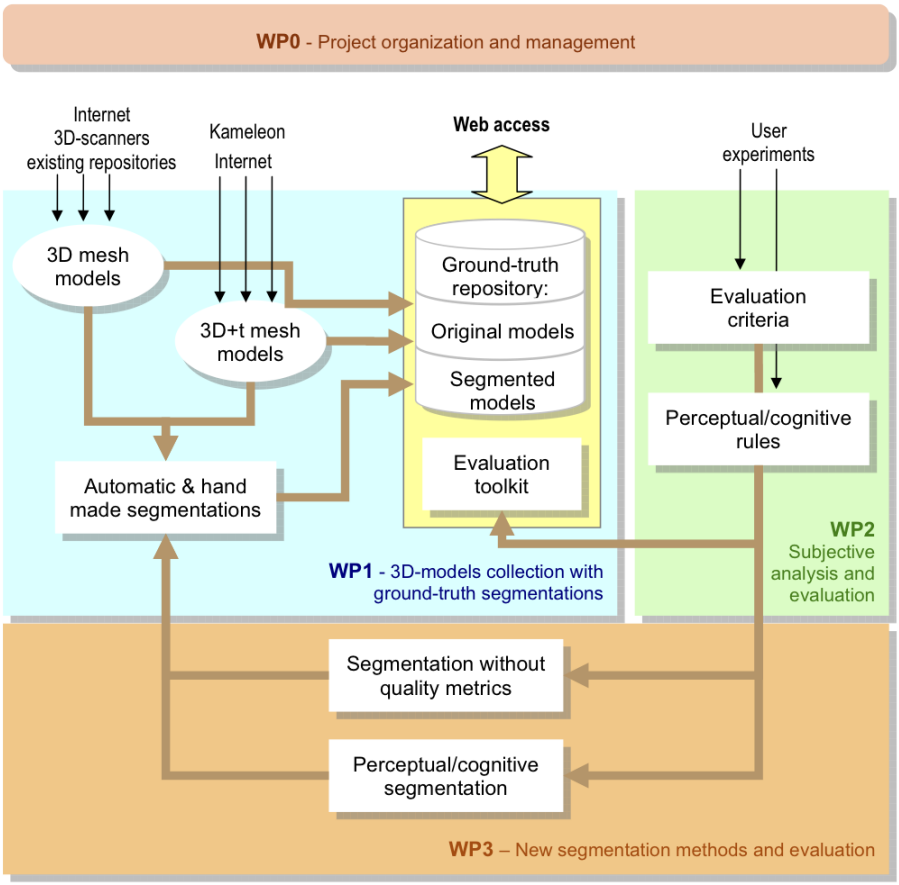


FIGURE 1.4 – *Vue d'ensemble des différentes tâches du projet et des liens entre les sous-projets.*

l'indexation partielle. L'indexation "traditionnelle" tend à considérer les modèles 3D en tant qu'objet unique et entier, que ce soit pour l'indexation off-line ou pour l'expression des requêtes. L'indexation partielle offre une plus grande souplesse en considérant les objets 3D comme des assemblages de parties permettant ainsi des recherches exprimées à partir d'un sous-ensemble d'un objet-requête. On peut, par exemple, exprimer la requête sous la forme d'une main et retrouver ainsi les modèles de mains présents dans la collection de données 3D, mais également tous les modèles dont une ou plusieurs parties sont similaires à la main requête (par exemple, un humanoïde complet).

Dans ce contexte, j'ai obtenu une bourse de thèse de l'Institut Télécom dans le cadre du programme *Futur et Ruptures*. Cette thèse est le travail de Rachid El Khoury et a commencé en novembre 2009.

Parallèlement, j'ai également co-encadré un autre doctorant, Hedi Tabia, bourse MESR. La thèse était dirigée par Mohamed Daoudi et Olivier Colot. Hedi Tabia a soutenu sa thèse le 27 septembre 2011.

Ce sujet est l'objet de la conclusion et des perspectives de mes travaux de recherche dans le présent document.

1.3.3 Encadrements

Doctorants

J'ai co-encadré, avec Mohamed Daoudi, trois thèses qui ont été soutenues :

- Tarik Filali Ansary, *Recherche de modèles tridimensionnels à partir de vues 2D*, thèse financée par le projet RNRT SEMANTIC-3D, **taux d'encadrement : 40%**. Thèse soutenue le 6 octobre 2006 devant le jury composé de M. Jean-Marc Geib (Professeur Université Lille 1, président du jury), M. Alberto Del Bimbo (Professeur Université de Florence, Italie, rapporteur), M. Remco Veltkamp (Professeur Université d'Utrecht, Pays-Bas, rapporteur), M. Atilla Baskurt (Professeur INSA Lyon, examinateur), M. Francis Schmitt (Professeur Télécom Paris Tech, examinateur), M. Mohamed Daoudi (Professeur Institut Télécom, directeur de thèse) et M. Jean-Philippe Vandeborre (Maître de Conférences Institut Télécom, co-encadrant de thèse).
  Tarik Filali Ansary est actuellement directeur adjoint à la direction

des forfaits et TMA de la société de services en informatique parisienne Normasys.

– Julien Tierny, *Reeb graph based 3D shape modeling and applications*, thèse financée par une allocation Ministère de l'Enseignement Supérieur et de la Recherche (MESR), **taux d'encadrement : 70%**. Thèse soutenue le 2 octobre 2008 devant le jury composé de Mme Sophie Tison (Professeur Université Lille 1, présidente du jury), M. Atilla Baskurt (Professeur INSA Lyon, rapporteur), M. Bruno Lévy (Directeur de Recherche INRIA Nancy Grand Est, rapporteur), M. Claude Labit (Directeur de Recherche INRIA Rennes, examinateur), M. Anuj Srivastava (Professeur Florida State University, USA, examinateur), M. Mohamed Daoudi (Professeur Institut Télécom, directeur de thèse) et M. Jean-Philippe Vandeborre (Maître de Conférences Institut Télécom, co-encadrant de thèse).
  Julien Tierny, après deux ans de post-doc à l'Université d'Utah, États-Unis, est actuellement chargé de recherches CNRS au LTCI / Télécom ParisTech, France.
– Halim Benhabiles, *3D-mesh segmentation : automatic evaluations and a new learning-based method*, en collaboration avec Guillaume Lavoué (LIRIS / INSA Lyon), thèse financée par le projet ANR MADRAS, **taux d'encadrement : 80%**. Thèse soutenue le 18 octobre 2011 devant le jury composé de M. Philippe Preux (Professeur Université Lille 3, président du jury), M. Christophe Rosenberger (Professeur ENSICAEN, rapporteur), Mme Michela Spagnuolo (Senior Researcher IMATI-GE / CNR, Italie, rapporteur), Florent Dupont (Professeur Université Lyon 1, examinateur), M. François Goulette (Professeur MINES ParisTech, examinateur), M. Mohamed Daoudi (Professeur Institut Télécom, directeur de thèse), M. Guillaume Lavoué (Maître de Conférences INSA Lyon, co-encadrant de thèse) et M. Jean-Philippe Vandeborre (Maître de Conférences Institut Télécom, co-encadrant de thèse).
  Halim Benhabiles est actuellement post-doctorant au laboratoire Le2i, Université de Bourgogne, France.

J'ai également collaboré quotidiennement avec un doctorant dirigé par Mohamed Daoudi et Olivier Colot (LAGIS, Université Lille 1) :

– Hedi Tabia, *Contributions to 3D-shape matching, retrieval and classification*, thèse financée par une allocation Ministère de l'Enseignement Supérieur et de la Recherche (MESR), soutenue le 27 septembre 2011

devant le jury composé de M. Denis Pellerin (Professeur Université Joseph Fourier, Grenoble, président du jury), M. Atilla Baskurt (Professeur INSA Lyon, rapporteur), M. Alberto Del Bimbo (Professeur Université de Florence, Italie, rapporteur), M. Michael M. Bronstein (Assistant-professor Università della Svizzera Italiana Lugano, Suisse, examinateur) et M. Jean-Philippe Vandeborre (Maître de Conférences Institut Télécom, examinateur).
Hedi Tabia est actuellement post-doctorant à l'Institut d'Electronique Fondamentale de l'Université ParisSud, Orsay, France.

Enfin, je co-encadre actuellement, toujours avec Mohamed Daoudi, une thèse en cours :

– Rachid El Khoury, *Indexation de modèles 3D par parties*, thèse commencée en novembre 2009, financée par le programme Futur et Ruptures de l'Institut Télécom. **Taux d'encadrement : 90%**.

Etudiants de Master Recherche (ou DEA)

J'ai encadré cinq étudiants en Master Recherche (ou DEA) d'Informatique de l'Université Lille 1 :

– Halim Benhabiles, *Contribution à l'évaluation automatique des méthodes de segmentation de maillages 3D polygonaux*, soutenu en juin 2008 ;
– Julien Tierny, *Hiérarchisation topologique de maillages de polygones – Une nouvelle méthode de construction de graphes de Reeb*, soutenu en juin 2005 ;
– Alexis Héloir, *Indexation de pièces automobiles par graphes de Reeb multirésolution*, soutenu en juin 2004 ;
– Adrien Thetten, *Recherche des vues caractéristiques à l'aide d'enveloppes visuelles*, soutenu en juin 2004 ;
– Tarik Filali Ansary, *Indexation et reconnaissance de modèles 3D de pièces mécaniques par une approche bayésienne*, soutenu en juin 2003.

Tous ces étudiants ont poursuivi leurs études en thèse dans notre équipe ou dans une autre université.

Autres stagiaires

Outre les encadrements en thèse et Master/DEA, j'ai régulièrement proposé et encadré des sujets de recherche aux étudiants de Télécom Lille 1 (dans le cadre des Projets Scientifiques et Technologiques) ainsi que dans

d'autres formations de l'Université Lille 1 (Maîtrise/Master1 d'Informatique, IUP Génie Mathématique et Informatique, DESS/MasterPro Genie Logiciel, etc.)

1.3.4 Participation à la communauté scientifique

En tant qu'enseignant-chercheur, je participe bien sûr à la vie de la communauté scientifique par mes relectures pour des revues et des conférences et ma présence dans des comités de programme de conférences.

Conférences et revues

- Relecteur pour les revues internationales Pattern Recognition, Signal Processing : Image Communication, The Visual Computer – International Journal of Computer Graphics, International Journal of Computer Vision, et SPIE Journal of Optical Engineering ;
- Membre des comités de programme pour les conférences et workshops International Workshops on Image Processing Theory, Tools and Applications (IPTA) depuis 2008, Eurographics Worshop on 3D Object Retrieval (EG-3DOR) depuis 2010, ACM Multimedia Workshop on 3D Object Retrieval (ACM-3DOR) depuis 2010, International ACM Workshop on Multimedia Access to 3D Human Objects (MA3HO) depuis 2011, CORESA (COmpression et REprésentation des Signaux Audiovisuels) depuis 2004 ;
- Co-organisateur des conférences CORESA'2004 et CORESA'2012 à Lille.
- Relecteurs pour diverses conférences (IEEE ICAR 2012, etc.) et journées de travail (Association Française d'Informatique Graphique, etc.)

Divers

Je suis membre de l'IEEE depuis plusieurs années, et membre du Groupe de Recherche Information, Signal, Images et viSion (GdR ISIS) dans le thème D – Télécommunications : compression, protection, transmission depuis 2001. Je suis également correspondant du LIFL au GdR ISIS depuis 2006.

1.4 Publications

L'ensemble de publications dont je suis co-auteur.

Nombre de publications de chaque type

Type de publication	Nombre
Chapitre de livre	1
Revues internationales de rang A	7
Conférences internationales avec comité de lecture	24
Conférences invitées	3
Conférences nationales avec comité de lecture	12
Rapports techniques et livrables liés à des projets	7
Autres publications (journées de travail, etc.)	7

Chapitre de livre

1. Stefano Berretti, Mohamed Daoudi, Alberto del Bimbo, Tarik Filali Ansary, Pietro Pala, Julien Tierny, and Jean-Philippe Vandeborre. *3D Object Processing : Indexing, Compression and Watermarking*, chapter 3, pages 87–138. John Wiley & Sons, April 2008. Book editors : Jean-Luc Dugelay, Atilla Baskurt and Mohamed Daoudi.

Revues internationales avec comité de lecture

1. Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. Learning boundary edges for 3d-mesh segmentation. *Computer Graphics Forum - Eurographics Association*, 30(8) :2170–2182, December 2011.
2. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. A new 3D-matching method of non-rigid and partially similar models using curve analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4) :852–858, April 2011.
3. Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. A comparative study of existing metrics for 3D-mesh segmentation evaluation. *The Visual Computer - International Journal of Computer Graphics*, 26(12) :1451–1466, December 2010.

4. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Partial 3D shape retrieval by Reeb pattern unfolding. *Computer Graphics Forum*, 28(1) :41–55, March 2009.
5. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer - International Journal of Computer Graphics*, 24(3) :155–172, March 2008.
6. Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. A bayesian 3D search engine using adaptive views clustering. *IEEE Transactions on Multimedia*, 9(1) :78–88, January 2007.
7. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. A framework for 3D CAD models retrieval from 2D images. *Annals of Telecommunications, special issue on Technologies and tools for 3D imaging*, 60(11-12) :1337–1359, November-December 2005.

Conférences et workshops internationaux avec comité de lecture

1. Rachid El Khoury, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3d mesh reeb graph computation using commute-time and diffusion distances. In *SPIE Electronic Imaging 2012 Symposium (3D Image Processing (3DIP) and Applications)*, volume 8290, Burlingame, California, USA, January 22-26 2012.
2. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. Non-rigid 3D shape classification using bag-of-feature techniques. In *IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, July 11-15 2011.
3. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. Deformable shape retrieval using bag-of-feature techniques. In *SPIE Electronic Imaging 2011 Symposium (3D Image Processing (3DIP) and Applications II (Conference 7864B))*, San Francisco, California, USA, January 23-27 2011.
4. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Collot. Local visual patch for 3D shape retrieval. In *ACM International Workshop on 3D Object Retrieval (in conjunction with ACM Multimedia 2010)*, Firenze, Italy, October 25 2010.
5. Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. A subjective experiment for 3D-mesh seg-

mentation evaluation. In *IEEE International Workshop on Multimedia Signal Processing (MMSP'2010)*, Saint-Malo, France, October 4-6 2010.

6. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Collot. 3D-shape retrieval using curves and HMM. In *20th IEEE International Conference on Pattern Recognition (ICPR 2010)*, Istanbul, Turkey, August 23-26 2010.
7. Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. In *IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2009)*, Beijing, China, June 26-28 2009. short paper.
8. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Fast and precise kinematic skeleton extraction of 3D dynamic meshes. In *19th IEEE International Conference on Pattern Recognition (ICPR-2008)*, Tampa, Florida, USA, December 8-11 2008.
9. Adrien Theetten, Tarik Filali Ansary, and Jean-Philippe Vandeborre. 3D-model view characterization using equilibrium planes. In *4th International Symposium on 3D Data Processing Visualization Transmission (3DPVT'08)*, Atlanta, Georgia, USA, June 18-20 2008.
10. Mohamed Daoudi, Tarik Filali Ansary, Julien Tierny, and Jean-Philippe Vandeborre. 3D-mesh models : view-based indexing and structural analysis. In *Digital Libraries : Research and Development. First International DELOS Conference*, volume LNCS 4877 of *Lecture Notes in Computer Science*, pages 298–307. Springer-Verlag, November 2007.
11. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Reeb chart unfolding based 3D shape signatures. In *Eurographics*, Prague, Czech Republic, September 3-7 2007. short paper.
12. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3D-model retrieval for mobiles. In *ACM International Conference on Image and Video Retrieval (CIVR 2007)*, Amsterdam, The Netherlands, July 9-11 2007. Technical Demonstration Session.
13. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Topology driven 3D mesh hierarchical segmentation. In *IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2007)*, Lyon, France, June 13-15 2007. short paper.

14. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. 3D mesh skeleton extraction using topological and geometrical analyses. In *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, Taipei, Taiwan, October 11-13 2006.
15. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. On 3D retrieval from photos. In *3rd IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA, June 14-16 2006.
16. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Invariant high level Reeb graphs of 3D polygonal meshes. In *3rd IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA, June 14-16 2006.
17. Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. 3D model retrieval based on adaptive views clustering. In *3rd International Conference on Advances in Pattern Recognition (ICAPR 2005)*, Bath, United Kingdom, August, 22-25 2005. Published in Lecture Notes in Computer Science by Springer-Verlag.
18. Chafik Samir, Jean-Philippe Vandeborre, and Mohamed Daoudi. Automatic 3D face recognition using topological techniques. In *IEEE International Conference on Multimedia & Expo (ICME 2005)*, Amsterdam, The Netherlands, July, 6-8 2005.
19. Adrien Theetten, Jean-Philippe Vandeborre, and Mohamed Daoudi. Determining characteristic views of a 3D object by visual hulls and Hausdorff distance. In *5th IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, Ottawa, Ontario, Canada, June, 13-17 2005.
20. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. A bayesian framework for 3D models retrieval based on characteristic views. In *2nd IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'04)*, Thessaloniki, Greece, September 6-9 2004.
21. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. A bayesian approach for 3D models retrieval based on characteristic views. In *17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, Cambridge, United Kingdom, August 23-26 2004.
22. Jean-Philippe Vandeborre, Vincent Couillet, and Mohamed Daoudi. A practical approach for 3D model indexing by combining local and

global invariants. In *1st IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'02)*, Padova, Italy, June, 19-21 2002.

23. Olivier Croquette, Jean-Philippe Vandeborre, Mohamed Daoudi, and Christophe Chaillou. Indexing and retrieval vrml models. In *SPIE Electronic Imaging 2002 Symposium (Internet Imaging III Conference)*, volume 4672, pages 95–106, San José, California, USA, January 21-25 2002.
24. Jean-Philippe Vandeborre, Mohamed Daoudi, and Christophe Chaillou. 3D models recognition from a 2D sketch. In *CESA'98/IEEE IMACS Multiconference*, Nabeul-Hammamet, Tunisia, April 1-4 1998.

Conférences invitées

1. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Geometry flavored topological skeletons : applications to shape handling, segmentation and retrieval. In *Second DELOS Conference on Digital Libraries*, Pisa, Italy, 5-7 December 2007.
2. Mohamed Daoudi, Tarik Filali Ansary, Julien Tierny, and Jean-Philippe Vandeborre. 3D-mesh models : view-based indexing and structural analysis. In *DELOS Conference 2007 Working Notes*, Pisa, Italy, February 13-14 2007.
3. Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. A bayesian 3D search engine using adaptive views clustering. In *Second IEEE-EURASIP International Symposium on Communications, Control and Signal Processing (ISCCSP 2006)*, Marrakech, Morocco, March 13-15 2006. First Special session : Content-based Access and Retrieval from Visual Digital Libraries (organized by Pr. Alberto del Bimbo, University of Firenze, Italy).

Conférences nationales avec comité de lecture

1. Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. Une expérimentation subjective pour l'évaluation de segmentations de maillages 3D. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2010)*, Lyon, France, 26-27 octobre 2010.
2. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Collot. Une approche pour la catégorisation des objets 3D basée

sur la théorie des fonctions de croyance. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2010)*, Lyon, France, 26-27 octobre 2010.

3. Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. Une collection de modèles 3D avec vérité-terrain pour l'évaluation objective des algorithmes de segmentation. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2009)*, Toulouse, France, 18-19 mars 2009.
4. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Signatures de formes 3D par dépliage de cartes de reeb. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2007)*, Montpellier, France, 8-9 novembre 2007.
5. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Graphes de Reeb de haut niveau de maillages polygonaux 3D. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2006)*, Caen, France, 9-10 novembre 2006.
6. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. Recherche de modèles 3D de pièces mécaniques basée sur les moments de Zernike. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2005)*, Rennes, France, 7-8 novembre 2005.
7. Chafik Samir, Jean-Philippe Vandeborre, and Mohamed Daoudi. Reconnaissance des visages 3D par les graphes de Reeb. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2005)*, Rennes, France, 7-8 novembre 2005.
8. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. Une approche bayésienne pour l'indexation de modèles 3D basée sur des vues caractéristiques. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2004)*, Lille, France, 25-26 mai 2004.
9. A. Baskurt, F. Blum, M. Daoudi, J.L. Dugelay, F. Dupont, A. Dutartre, T. Filali Ansary, F. Fratani, E. Garcia, G. Lavoué, D. Lichau, F. Prêteux, J. Ricard, B. Savage, J.P. Vandeborre, and T. Zaharia. SEMANTIC-3D : compression, indexation et tatouage de données 3D. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2004)*, Lille, France, 25-26 mai 2004.
10. Sadiq Abdelalim, Rachid Oulad Haj Thami, Mohamed Daoudi, and Jean-Philippe Vandeborre. Classification des objets 3D basée sur la

logique floue. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2004)*, Lille, France, 25-26 mai 2004.

11. Jean-Philippe Vandeborre, Mohamed Daoudi, Alain Preux, and Christophe Chaillou. Recalage automatique d'un modèle d'objet et récupération de textures sur des photos quelconques. In *Compression et Représentation des Signaux Audiovisuels (CORESA'2000)*, Poitiers-Futuroscope, France, 19-20 octobre 2000.
12. Jean-Philippe Vandeborre, Mohamed Daoudi, Alain Preux, and Christophe Chaillou. Adaptation d'un modèle générique d'objet et récupération de textures à partir de photos quelconques. In *Compression et Représentation des Signaux Audiovisuels (CORESA'99)*, Sophia-Antipolis, France, 14-15 juin 1999.

Rapports techniques

1. Julien Tierny, Tarik Filali Ansary, and Jean-Philippe Vandeborre. A novel method for constructing meaningful Reeb graphs from piecewise continuous functions over 2-manifolds. Technical Report 2005-05, Laboratoire d'Informatique Fondamentale de Lille (LIFL - UMR USTL/CNRS 8022), December 2005.
2. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. Une approche bayésienne pour l'indexation de modèles 3D basée sur des vues caractéristiques. Technical Report 2004-07, Laboratoire d'Informatique Fondamentale de Lille (LIFL - UMR USTL/CNRS 8022), 2004.

Rapports techniques liés à des projets

1. Halim Benhabiles, Guillaume Lavoué, and Jean-Philippe Vandeborre. Result analysis and evaluation criteria proposition for objective evaluation of mesh segmentation algorithm. Technical Report D2.3, Projet ANR MADRAS, December 2009.
2. Halim Benhabiles, Guillaume Lavoué, and Jean-Philippe Vandeborre. Protocol definition and experiment for subjective analysis and evaluation of mesh segmentation algorithm. Technical Report D2.2, Projet ANR MADRAS, December 2009.
3. Romain Arcila, Halim Benhabiles, Mohamed Daoudi, Florent Dupont, Franck Hétroy, and Jean-Philippe Vandeborre. Segmentation

de maillages et de séquences de maillages : un état de l'art. Technical Report D3.1, Projet ANR MADRAS, October 2008.

4. Mohamed Daoudi, Tarik Filali Ansary, Julien Tierny, Jean-Philippe Vandeborre, and 9 other authors. Delos research activities 2006. Technical report, DELOS : a Network of Excellence on Digital Libraries, August 2006.
5. Titus Zaharia, Françoise Prêteux, Julien Ricard, Florent Dupont, Atilla Baskurt, Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. Critères pour la comparaison des méthodes d'indexation. Technical Report L2.4, Projet RNRT 2002 SEMANTIC-3D, juin 2005.
6. Atilla Baskurt, Mohamed Daoudi, Tarik Filali Ansary, Julien Ricard, and Jean-Philippe Vandeborre. Evaluation et comparaison des méthodes de recherche d'objets 3D à partir de données 2D. Technical Report L2.3, Projet RNRT 2002 SEMANTIC-3D, juin 2004.
7. Atilla Baskurt, Mohamed Daoudi, Tarik Filali Ansary, Saïd Mahmoudi, Françoise Prêteux, Julien Ricard, Jean-Philippe Vandeborre, and Titus Zaharia. Etat de l'art sur l'indexation 3D. Technical Report L2.1, Projet RNRT 2002 SEMANTIC-3D, juin 2003.

Autres publications (journées de travail, workshop, etc.)

1. Halim Benhabiles, Guillaume Lavoué, and Jean-Philippe Vandeborre. Une nouvelle métrique de similarité pour l'évaluation de segmentations de maillages 3D en utilisant un corpus de vérités-terrains. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, Arles, France, 25-27 novembre 2009.
2. Tarik Filali Ansary, Jean-Philippe Vandeborre, and Mohamed Daoudi. Recherche d'objets 3D à partir de photos. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, Bordeaux, France, 22-24 novembre 2006.
3. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. Analyse topologique et géométrique de maillages 3D pour l'extraction de squelette. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, Bordeaux, France, 22-24 novembre 2006.
4. S. Berretti, A. Del Bimbo, P. Pala, M. Chaouch, R. Cucchiara A. Verroust, C. Grana, F. Seinstra, M. Worring, M. Daoudi, and J.-P. Vandeborre. Description, matching and retrieval by content of 3D objects.

In *DELOS proceedings, in conjunction with the European Conference on Research and Advanced Technology for Digital Libraries (ECDL'06)*, Alicante, Spain, September 2006.

5. Jacques Landru and Jean-Philippe Vandeborre. VIMINAL (VIrtual Model for Ip Network Architecture Lab). In *Journées Réseaux JRES 2005*, Marseille, France, 5-9 décembre 2005.
6. Alexis Héloir, Chafik Samir, Jean-Philippe Vandeborre, and Mohamed Daoudi. Indexation de pièces automobile par graphe de Reeb multirésolution. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, pages 218–225, 24-26 novembre 2004.
7. Jean-Philippe Vandeborre, Mohamed Daoudi, Alain Preux, and Christophe Chaillou. Etude sur l'extraction de textures à partir de photos quelconques. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, Dunkerque, France, 2-4 December 1998.

Indexation de modèles 3D

2

Comme cela a été souligné en introduction de ce document, l'utilisation de modèles tridimensionnels dans les données multimédia est en forte croissance. Avec l'apparition de scanners 3D et la montée en puissance et en simplicité des logiciels de création 3D, il devient de plus en plus aisé de créer un modèle 3D très réaliste et détaillé. Une question que se posent alors les concepteurs de mondes virtuels est « Comment puis-je exploiter au mieux les immenses collections de modèles 3D existantes ? Comment faire des recherches dans ces collections le plus efficacement et le plus simplement possible ? »

Pour répondre à cette question, la solution qui vient immédiatement à l'esprit, et qui a déjà été utilisée pour indexer des images (Google Images par exemple), est l'indexation textuelle des collections : décrire chaque élément de la collection avec un ou plusieurs mots-clefs. Cette solution, aussi simple qu'elle paraisse à première vue, présente de nombreux inconvénients. Pour n'en citer que deux, il y a tout d'abord le temps nécessaire à l'étiquetage (*tag* ou *labeling*) de la collection, et l'aspect subjectif des mots-clefs employés, ces mots-clefs étant nécessairement liés à la perception et à la culture de l'opérateur qui effectue l'étiquetage.

La méthode la plus efficace, et certainement la plus intuitive pour l'utilisateur final, est connue sous le terme de *recherche par l'exemple*. Elle consiste à caractériser les modèles 3D grâce à des descripteurs de forme invariants à certaines transformations géométriques – notamment translations, rotations et homothéties. L'utilisateur exprime alors sa requête sous la forme d'un exemple de ce qu'il recherche dans la collection. La comparaison entre les descripteurs de la requête et ceux des modèles dans la collection se fait grâce à une métrique. Pour comparer la requête aux modèles de la collection, la métrique doit donc exprimer une distance qui se traduise en une "distance visuelle" entre deux objets.

Notons ici que nous nous intéressons uniquement à la forme des objets. Les couleurs et/ou textures associées aux illustrations présentées dans ce chapitre sont purement décoratives. Elles n'ont aucune pertinence ni aucun impact sur les résultats exposés dans ce chapitre et le reste du mémoire.

Dans ce chapitre, nous présentons les recherches que nous avons menées avec l'encadrement de la thèse de Tarik Filali Ansary soutenue le 6 octobre 2006. Nous commençons par exposer notre approche de l'indexation par vues caractéristiques. Quelques expériences montrent ensuite la justesse des choix que nous avons faits et positionnent notre méthode par rapport à d'autres méthodes reconnues. Enfin, le moteur de recherche 3D pour le web que nous avons développé est présenté.

2.1 Notre approche par vues caractéristiques

Plusieurs raisons nous ont poussé à proposer une méthode utilisant des vues 2D pour caractériser un objet 3D.

Le système de vision humain a une capacité forte à reconnaître les objets tridimensionnels grâce à une seule et unique image bi-dimensionnelle. D'ailleurs, la question de savoir si la reconnaissance d'objets 3D repose sur une représentation intellectuelle purement tridimensionnelle ou sur un ensemble de vues bi-dimensionnelles a été étudiée notamment par Riesenhuber et Poggio (2000). Ils ont montré que, dans le système de vision et de reconnaissance humain, un objet 3D est représenté principalement par un ensemble de vues 2D qui caractérise l'objet 3D.

De nombreux états de l'art comme celui de Tangelder et Veltkamp (2004) et des ouvrages spécialisés en 3D comme celui de Dugelay et al. (2008, chapitre 2), montrent que les méthodes orientées vues sont très efficaces. Le *Princeton Shape Benchmark*[1] (Shilane et al. 2004) donne des comparaisons entre différentes méthodes d'indexation 3D que nous utiliserons également en section 2.2 pour souligner les performances de notre approche. Il classe les méthodes orientées vues parmi les meilleures.

De plus, l'indexation de collections d'objets 3D de mauvaise qualité (basse résolution, orientation des facettes pas toujours cohérente, trous dans le maillage, maillage non manifold, etc.) contraint à utiliser un descripteur qui ne soit pas trop sensible à la mauvaise qualité des maillages.

1. http://shape.cs.princeton.edu/benchmark/

A partir du moment où il est possible d'obtenir des vues d'un maillage 3D – même si celui-ci n'est pas d'excellente qualité, les méthodes orientées vues sont tout à fait utilisables et adaptées.

Enfin, la comparaison de deux objets 3D par une méthode orientée vues est très simple et intuitive. Elle s'appuie sur le fait que si différents points de vue de deux modèles 3D sont similaires entre eux, alors les deux modèles 3D sont similaires en tout point. La méthode que nous proposons consiste donc à faire correspondre une ou plusieurs vues (croquis, photos, etc.) aux modèles 3D similaires.

2.1.1 Présentation rapide de la méthode

Comme souligné précédemment, l'idée principale de l'indexation 3D à partir de vues est que deux modèles 3D sont similaires s'ils se ressemblent aussi de différents points de vue. Ceci nous conduit à penser l'interrogation d'une base de modèles 3D comme l'expression de la requête sous la forme d'une ou plusieurs vues 2D (photo, croquis, image, etc.) montrant l'objet recherché sous différents points de vue.

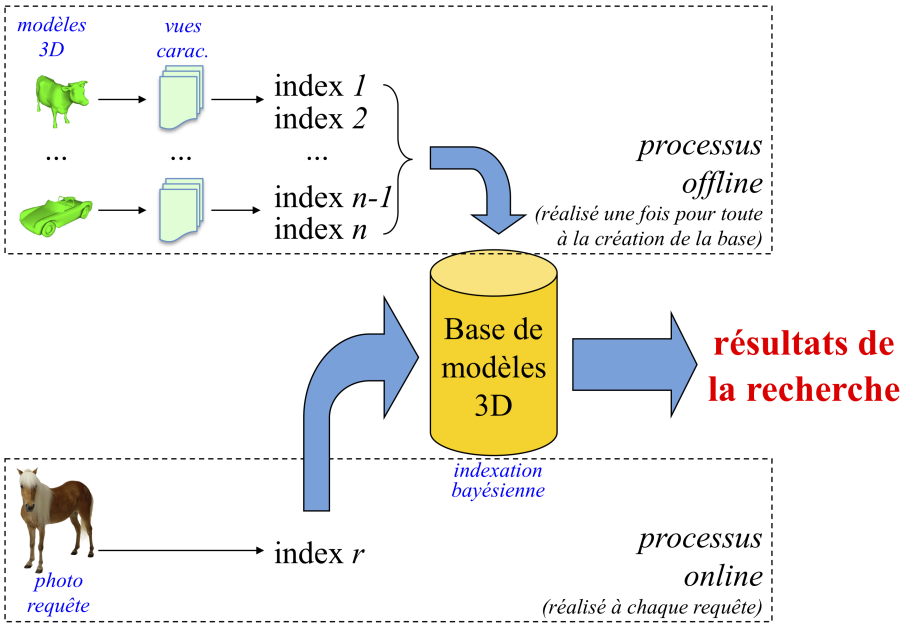


FIGURE 2.1 – *L'indexation et la recherche de modèles 3D par vues 2D.*

L'indexation et la recherche de modèles 3D en utilisant des vues 2D se décompose en les deux étapes suivantes, illustrées par la figure 2.1 :

– Indexation de la base de modèles 3D (processus *offline*) : chaque modèle 3D de la base est caractérisé par un ensemble de vues 2D. Cet ensemble de vues forme ce qu'on appelle les *vues caractéristiques* de

l'objet 3D. Ensuite, pour chaque vue caractéristique, un descripteur 2D est calculé.

– Recherche de modèles 3D (processus *online*) : le même descripteur que celui utilisée à l'étape précédente est calculé sur la ou les vues requêtes. Un processus de recherche est alors lancé pour trouver les vues caractéristiques qui correspondent le plus aux vues requêtes, et par conséquent aux modèles 3D les plus proches visuellement.

2.1.2 Descripteur de vues

Le descripteur de vues 2D utilisé doit respecter des propriétés mathématiques (invariance aux transformations géométriques de translation, rotation et mise à l'échelle) et être facilement calculable.

De nombreux descripteurs sont alors possibles. Nous en avons étudié trois : les histogrammes de courbures, le descripteur CSS (*Curvature Scale Space*) et les moments de Zernike. Finalement, ce sont les moments de Zernike qui ont donné les meilleurs résultats.

Histogramme de courbures

Les histogrammes de courbures utilisent un descripteur de forme appelé *curvature index* et introduit par Koenderink et van Doorn (1992). Cet index de courbures a été utilisé pour la reconnaissance d'images (Nastar 1997) ou comme descripteur local de maillages 3D (Zaharia et Prêteux 2001, Vandeborre et al. 2002) par exemple. Il est défini comme une fonction des courbures principales d'une surface en un point.

Pour utiliser ce descripteur pour les vues caractéristiques 2D, une vue est convertie en une image à niveaux de gris. La vue est alors considérée comme une surface tridimensionnelle avec le niveau de gris comme coordonnée d'élévation Z. L'index de courbures peut alors être calculé en chaque point de la surface et un histogramme de distribution de cet index est créé.

Chaque vue étant représentée par un histogramme de courbures, la comparaison de deux vues se fait simplement en comparant leurs histogrammes respectifs grâce à un distance comme celle de Minkowski L_1.

Nous avons publié les détails de ce descripteur dans la conférence *IEEE International Conference on Pattern Recognition (ICPR)* en 2004 (Filali Ansary et al. 2004a).

Descripteur CSS (*Curvature Scale Space*)

Le descripteur CSS (*Curvature Scale Space*) a été introduit par Mokhtarian et Mackworth (1986) pour la reconnaissance de formes 2D. Il a également été utilisé par Mahmoudi et Daoudi (2002) pour la reconnaissance de formes 3D à partir de vues.

Pour utiliser ce descripteur, il est nécessaire d'extraire le contour de l'objet représenté pour en faire une courbe fermée. Ceci se fait facilement à partir d'une image binarisée (en noir l'objet, en blanc le fond).

Moments de Zernike

Les moments de Zernike sont des moments orthogonaux invariants aux rotations. Ils sont construits en utilisant un ensemble de polynômes complexes qui forment un ensemble orthogonal complet défini sur le disque unité. Ils ont été inventés en 1934 et ont notamment été utilisés pour la reconnaissance d'images par Khotanzad et Hong (1990), qui en font d'ailleurs une description détaillée.

Les moments de Zernike sont invariants uniquement aux rotations. Pour obtenir l'invariance aux translations et aux mises à l'échelle, une vue 2D subit tout d'abord un processus de normalisation avant tout calcul de moments. Les moments s'appuient sur la notion d'intérieur/extérieur sur l'image : dans l'image binarisée, les pixels noirs appartiennent à l'objet, les pixels blancs appartiennent au fond.

Nous avons publié les détails de ces trois descripteurs (histogramme de courbures, CSS et moments de Zernike) ainsi qu'une comparaison de ceux-ci dans le cadre de l'utilisation des vues caractéristiques pour l'indexation 3D dans la conférence *IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT)* en 2004 (Filali Ansary et al. 2004b).

Choix du descripteur de vues

Nous avons mené une étude sur chacun de ces descripteurs en utilisant la base de pièces mécaniques fournie par le constructeur automobile Renault, partenaire du projet SEMANTIC-3D (voir chapitre 1 pour la des-

cription du projet). Quelques exemples de ces pièces mécaniques[2] sont visibles à la figure 2.2.

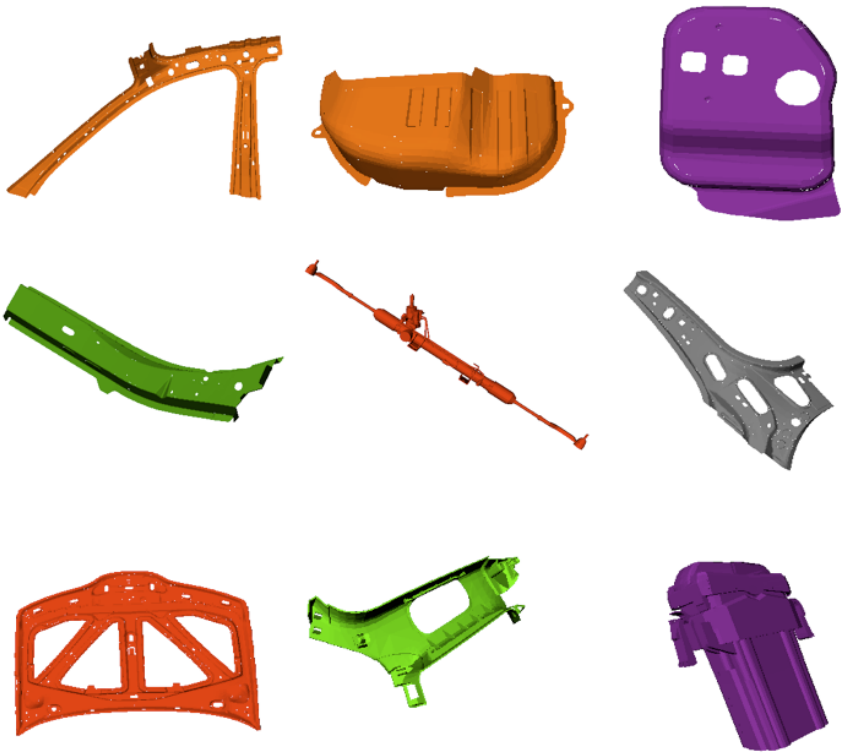


FIGURE 2.2 – *Quelques pièces mécaniques de la base fournie par Renault dans le cadre du projet SEMANTIC-3D.*

L'histogramme des courbures est calculé sur une vue et dépend donc de la position des sources de lumière qui ont servi à calculer le rendu de cette vue. Ce qui n'est pas gênant pour l'indexation à partir des vues des objets à indexer – si on fixe bien les positions des sources lumineuses de manière identique pour toutes les vues – mais cela le devient dans les requêtes exprimées par l'utilisateur pour lesquelles il est impossible de contrôler ces paramètres. L'histogramme des courbures n'est donc pas le candidat idéal pour décrire les vues dans notre approche de l'indexation de modèles 3D par vues.

Le descripteur CSS dépend quant à lui du contour extérieur de l'objet représenté sur une vue. Or, les pièces mécaniques de la base Renault présentent de nombreux trous intérieurs caractéristiques des objets (par exemple sur les différentes versions de conception d'une porte de la Laguna II illustrée à la figure 2.3) ainsi que de nombreux artefacts dus au

2. Pour rappel, qui a déjà été indiqué en introduction de ce chapitre, les couleurs et/ou textures associées aux illustrations présentées dans ce document sont purement décoratives. Elles n'ont aucun sens particulier ni aucun impact sur les résultats présentés.

passage du modèle Catia [3] au maillage (maillages non manifolds, irrégularités à la jointure des patches de maillages, etc.). Ceci fait que les vues d'une pièce mécanique sont très rarement constituées d'un seul contour continu et fermé. De plus, par leur conception et leur utilisation, les pièces mécaniques présentent généralement peu de formes courbes. Or, le descripteur CSS tire ses principales informations des courbes prononcées. Le descripteur CSS n'est donc pas non plus un bon candidat pour notre approche.

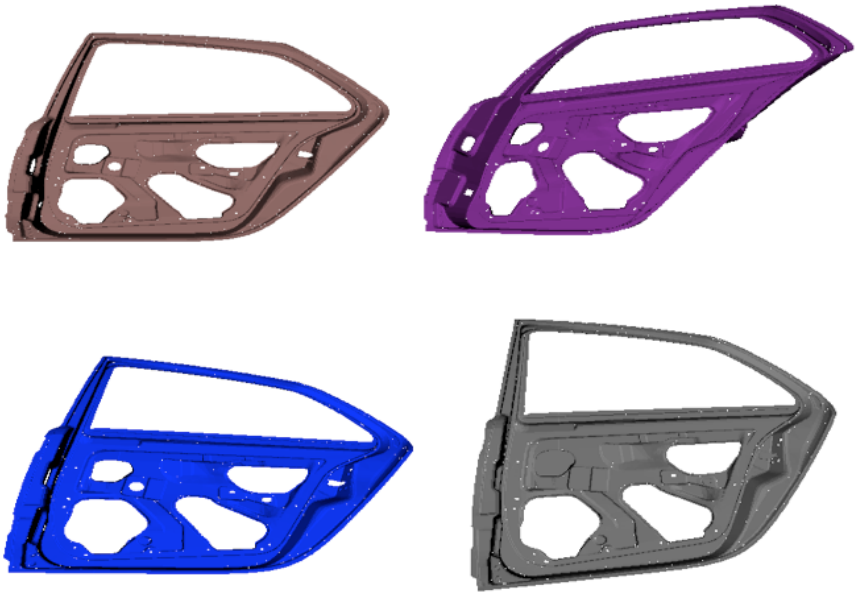


FIGURE 2.3 – *Différentes versions d'une porte de Laguna II de la base Renault.*

Les moments de Zernike sont un descripteur global d'une vue 2D. Ce descripteur s'applique en fait sur une image binaire de la vue et prend bien en compte les trous et leurs positions. Binariser une vue pendant l'étape *offline* est très simple. Il suffit d'effectuer un rendu de la vue sans aucune source lumineuse. Dans l'étape *online*, tout dépend de la provenance de l'image-requête fournie par l'utilisateur. Mais généralement, la conversion n'est pas très compliquée. Un des inconvénients de ce descripteur est la présence de multiples objets se cachant les uns les autres. Il est alors impossible de distinguer les différents objets dans l'image binaire. Nous avons donc restreint l'utilisation des images-requêtes à des vues ne représentant qu'un seul objet, la distinction de multiples objets sur une image sortant du cadre de ces recherches. Un autre inconvénient est sa dépendance à la localisation de l'objet sur la vue. Pour résoudre ce pro-

3. Logiciel de modélisation 3D de Dassault Systèmes très utilisé pour la conception de pièces industrielles – `http://www.3ds.com/products/catia`

blème, nous avons mis en place une étape de pré-traitement de la vue pour la normaliser par rapport à toutes celles utilisées dans la base indexée.

Les moments de Zernike sont donc le candidat idéal pour notre approche de l'indexation de maillages 3D à partir de vues caractéristiques.

2.1.3 Sélection des vues caractéristiques

Pour indexer des modèles 3D à partir de vues, la première étape consiste naturellement à sélectionner les vues qui caractérisent chaque modèle 3D comme l'illustre la figure 2.4.

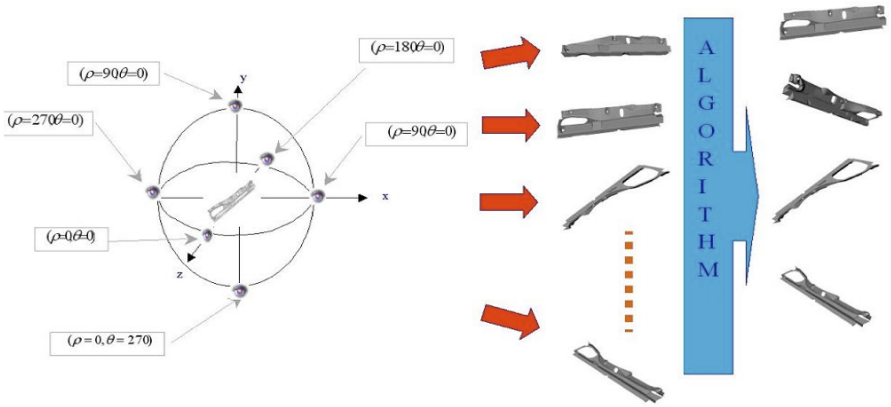


FIGURE 2.4 – *Sélection des vues caractéristiques d'un modèle 3D.*

La sélection de ces vues caractéristiques doit répondre à deux contraintes :

1. Nombre de vues. Le nombre de vues caractéristiques d'un objet 3D doit être suffisamment grand pour représenter tous les points de vue essentiels, et ainsi toute l'information nécessaire à la caractérisation de l'objet, mais sans être trop important pour ne pas obtenir une sur-représentation néfaste aux performance de l'approche.
2. Position des vues. Outre le nombre, chaque vue doit représenter un point de vue – *ie.* une position particulière de l'objet 3D – différent et apportant un maximum d'information à la caractérisation de l'objet 3D.

Pour satisfaire ces contraintes, nous avons développé un algorithme de sélection des vues caractéristiques qui choisit le nombre de vues selon la complexité de l'objet et la position de ces vues selon le degré d'information qu'elles apportent. En effet, selon la complexité de l'objet à caractériser, le

nombre de vues peut varier grandement. De plus, toutes les vues d'un objet n'ont pas la même importance, certaines apportant plus d'informations que d'autres. Par exemple, pour caractériser une sphère, une seule vue suffit puisque toutes les points de vues autour de cet objet sont strictement identiques. Par contre, la caractérisation d'un objet plus complexe – comme une pièce mécanique, un avion, un humanoïde, etc. – demande un plus grand nombre de vues et un choix judicieux des points de vue.

L'idée générale de l'algorithme est de générer un ensemble initial de vues (320 dans notre cas) puis de sélectionner uniquement celles qui caractérisent au mieux le modèle 3D en satisfaisant les contraintes selon un critère informationnel. La suite de ce paragraphe détaille les étapes de l'algorithme.

Création de l'ensemble initial de vues

Pour générer l'ensemble initial, des vues – *ie.* des projections du modèle 3D – sont créées à partir de nombreux points de vue. Ces points de vue sont répartis de manière équi-distante sur la sphère unitaire englobante de l'objet 3D. L'objet est mis à l'échelle et centré sur la sphère englobante. Pour créer cette sphère, nous partons d'un icosaèdre régulier qui est ensuite subdivisé jusqu'à obtenir un polyèdre régulier de 320 faces approximant une sphère. Une caméra est alors placée au centre de chaque face et pointe vers le centre de la sphère. Chaque caméra donne alors une vue par simple projection. Une vue est une simple silhouette. Les moments de Zernike sont ensuite extraits de chaque vue. Chaque vue est enfin décrite par Z coefficients des moments extraits.

Réduction de l'ensemble initial

La second étape consiste à réduire l'ensemble initial de 320 vues pour ne garder que celles représentant au mieux l'objet 3D.

Chaque vue de l'ensemble (320 vues) étant décrit par Z coefficients des moments de Zernike, trouver le sous-ensemble de vues qui caractérise au mieux l'objet 3D équivaut à trouver le sous-ensemble de points qui représente un ensemble de 320 points dans un espace à Z dimensions. Il s'agit là d'un problème très connu de classification.

La classification de données est un problème largement couvert par la littérature mathématique et informatique. De nombreux états de l'art

couvrent ce domaine, comme celui de Jain et al. (1999). On peut citer de nombreuses méthodes de classification comme la classification hiérarchique (*hierarchical clustering* – King (1967), la classification par plus proche voisin (*nearest neighbour clustering* – Lu et Fu (1978), la classification par partitions (*partitioning clustering* – Dubes (1987), la méthode *K-means* (Duda et Hart 1973) et ses dérivées comme les *K-medoids*.

Au début de nos travaux, nous nous sommes intéressés à plusieurs de ces algorithmes de classification. Nous avons par exemple publié une méthode adaptée de la classification hiérarchique (Filali Ansary et al. 2005) qui donne de bons résultats.

Cependant, la majorité de ces algorithmes, et notamment les méthodes de type K-means, nécessite de fixer à l'avance le fameux paramètre K alors que nous souhaitons que l'algorithme adapte le nombre de vues à la complexité de l'objet. Chaque objet est donc caractérisé par un nombre de vues différent.

AVC – *Adaptative Views Clustering*

Pour apporter une solution à ce problème, nous avons développé un algorithme que nous avons nommé *Adaptative Views Clustering* ou AVC. Cet algorithme est principalement adapté de la méthode *X-means* proposée par Pelleg et Moore (2000). La méthode X-means ne nécessite pas de fixer le fameux nombre de classes K imposé par les méthodes de type K-means. Cette méthode s'appuie néanmoins sur l'algorithme K-means et réitère ce dernier jusqu'à obtenir un partitionnement et un nombre K de partitions idéal selon un critère informationnel (comme le critère BIC (*Bayesian Information Criteria*) expliqué dans la section suivante). Le nombre K doit être borné en entrée de l'algorithme. Pour notre approche, nous avons choisi un K compris entre 1 et 40. Selon nos expériences, cet intervalle est un bon compromis entre vitesse d'exécution et précision de la représentation des modèles 3D par ses vues caractéristiques.

Pour résumer brièvement l'algorithme du K-means, un ensemble d'éléments – ici de l'ensemble initial des 320 vues d'un modèle 3D – est réparti en K partitions, K fixé en entrée de l'algorithme. Chaque partition est alors représentée par un élément qui est le centre de celle-ci. L'algorithme procède de manière itérative pour affiner le partitionnement jusqu'à ce que les représentants des partitions ne changent plus – à un ϵ près – d'une itération à l'autre. Il faut noter que cet algorithme est parfaitement dé-

terministe mais dépend grandement du choix des vues caractéristiques initiales. Le choix des vues initiales peut avoir un impact important sur les performances de l'algorithme et sur le résultat final.

L'algorithme X-means utilise les K-means de manière itérative en partant de $K = 1$ et en incrémentant K à chaque itération jusqu'à atteindre la borne supérieure définie (dans notre cas, $K = 40$). A chaque itération, un score informationnel est calculé. Finalement,le nombre de vues K présentant le meilleur score fixe le nombre optimal de vues pour caractériser un objet 3D.

BIC – *Bayesian Information Criteria*

Trouver un *modèle* qui représente au mieux un ensemble de données est un problème mathématique connu sous le nom de *sélection d'un modèle de représentation*. Dans notre cas, un modèle de représentation est un ensemble de vues caractéristiques.

Ce problème peut se résoudre en utilisant de nombreux critères informationnels comme AIC (*Akaike Information Criteria* – Akaike (1973)), BIC (*Bayesian Information Criteria* – Schwarz (1978)), *Bayes Factor* (Berger et Pericchi 1994), etc. Nos expériences et l'étude de Lebarbier et Mary-Huard (2004) nous ont conduit à choisir le critère BIC pour notre problème de sélection des vues caractéristiques.

Tous les détails concernant la sélection des vues caractéristiques d'un modèle 3D sont donnés dans la section II de la publication *A Bayesian 3D Search Engine using Adaptive Views Clustering* (Filali Ansary et al. 2007) reproduite en annexe A page 144 du présent document.

Un exemple de vues caractéristiques d'un modèle 3D

Nous donnons ici un exemple de vues caractéristiques sélectionnées pour un modèle 3D de type *voiture* appartenant à la collection d'objets 3D du *Princeton Shape Benchmark*.

La figure 2.5 illustre l'évolution du score BIC en fonction du nombre de vues caractéristiques de la voiture sur la partie gauche de l'illustration. Cette courbe montre que le score BIC maximal est atteint pour 17 vues caractéristiques. Ces 17 vues forment alors l'ensemble "optimal" des vues caractéristiques de cet objet 3D au sens de notre algorithme AVC.

La figure 2.6 montre cet ensemble de 17 vues caractéristiques.

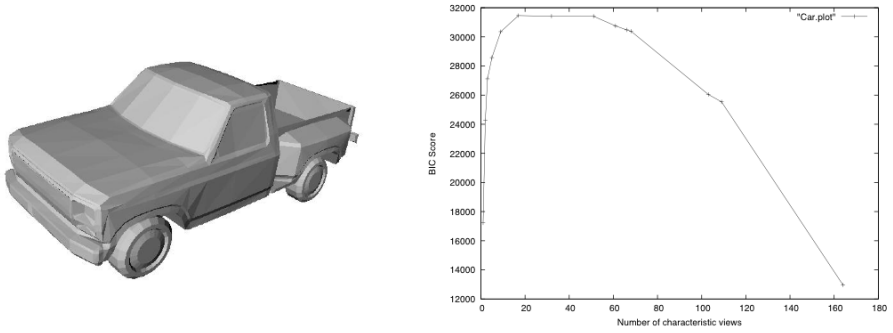


FIGURE 2.5 – *Un modèle 3D* voiture *et l'évolution du score BIC en fonction du nombre de vues caractéristiques.*

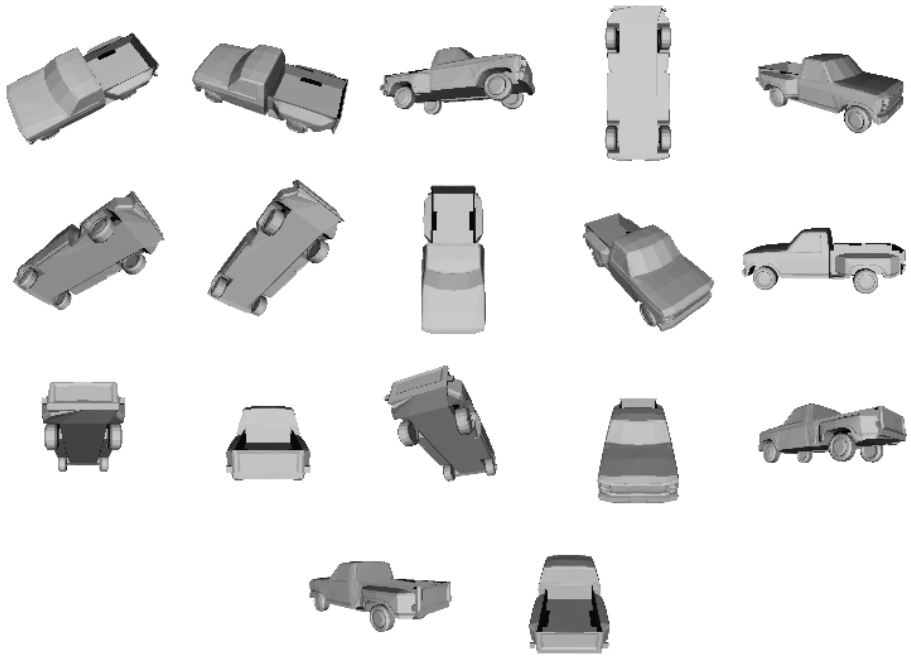


FIGURE 2.6 – *17 vues caractéristiques d'un modèle 3D* voiture.

2.1.4 Indexation probabiliste

Maintenant que chaque modèle de la collection à indexer est représenté par un nombre optimal de vues caractéristiques et que chacune de ces vues caractéristiques est décrite par un descripteur 2D (coefficients des moments de Zernike), nous nous intéressons à la partie *online* du système, à savoir la recherche de modèles 3D à partir d'une vue ou d'un modèle 3D requête.

Comme nous l'avons plusieurs fois soutenu dans ce chapitre, l'idée principale que nous défendons est que **deux objets 3D sont similaires**

s'ils se ressemblent également de différents points de vues. Autrement dit, la comparaison de plusieurs vues de modèles 3D et le rapprochement de celles-ci en terme de distance entre descripteurs de vues, va nous aider à déterminer les modèles 3D qui se rapprochent le plus de la requête.

Deux remarques essentielles sont à souligner. Tout d'abord, même si chaque modèle 3D est représenté par un ensemble optimal de vues caractéristiques, certaines vues portent en elles plus d'informations essentielles, plus de détails, que d'autres, dans le sens où elles sont plus susceptibles d'être représentées dans une requête. Ensuite, nous avons constaté durant notre travail que certains objets simples (comme un cube ou une sphère) peuvent être à la base d'objets plus complexes. La figure 2.7 montre par exemple plusieurs objets ayant une forme cubique à la base. Ces objets auront donc des vues caractéristiques communes comme celles montrant un simple carré.

FIGURE 2.7 – *Un cube et trois objets dérivés.*

Ces deux remarques nous ont donc amené à développer une méthode de recherche prenant en compte le fait que certaines vues caractéristiques représentaient mieux un objet que d'autres, et avaient donc une plus grande probabilité d'apparaître, ainsi que le fait que des objets simples ont également une plus forte probabilité d'apparaître dans les résultats de la recherche que les objets plus complexes. C'est donc fort logiquement que nous avons proposé une méthode probabiliste de la recherche de modèles 3D à partir de vues.

Tous les détails et développement mathématiques concernant notre méthode probabiliste bayésienne sont donnés dans la section III de la publication *A Bayesian 3D Search Engine using Adaptive Views Clustering* (Filali Ansary et al. 2007) reproduite en annexe A page 144 de ce mémoire.

2.2 Résultats et positionnement de notre approche

Nous présentons ici les résultats obtenus avec la méthode présentée dans les pages précédentes. Nous commençons tout d'abord par une pré-

sentation rapide des deux collections de données sur lesquelles nous avons travaillé : la base de pièces mécaniques Renault, utilisée dans le cadre du projet RNRT SEMANTIC-3D (plus de détails au chapitre 1 page 5), et la base du *Princeton Shape Benchmark*[4] (Shilane et al. 2004). Nous comparons ensuite les performances de notre approche par rapport aux méthodes développées par les partenaires du projet RNRT SEMANTIC-3D sur la base Renault, et par rapport à treize méthodes de l'état de l'art utilisant la base du *Princeton Shape Benchmark*. Enfin, quelques résultats utilisant des photographies réelles sont montrés.

2.2.1 Collection de données

Pour évaluer les performances de notre méthode, nous avons utilisé deux collections de données : la base Renault pour les besoins du projet RNRT SEMANTIC-3D et, puisque la base Renault est par nature privée et protégée par ce dernier, la base du *Princeton Shape Benchmark*, incontournable dans le domaine de l'indexation de modèles 3D.

Base Renault du projet RNRT SEMANTIC-3D

La base Renault a été conçue pour les besoins du projet RNRT SEMANTIC-3D et pour permettre aux partenaires du projet de concevoir des méthodes de compression, d'indexation et de tatouage de données 3D. Elle est constituée de modèles de pièces de la voiture Renault Laguna II, et plus particulièrement de 5000 pièces mécaniques de l'avant-droit du véhicule. Pour évaluer les performances des méthodes, nous avons sélectionné, avec l'aide des autres partenaires du projet, 758 pièces représentatives pour former notre base de test. Ces 758 pièces ont été réparties en 75 classes (portières, phares, ailes, etc.), selon le jugement de deux personnes et la nomenclature des pièces Renault. La figure 2.2 page 40 donne quelques exemples de ces pièces mécaniques. Notons enfin que Renault utilise le logiciel de modélisation Catia pour élaborer ses prototypes de pièces. Les modèles 3D fournis sont de simples maillages obtenus à partir de triangulation de surfaces NURBS notamment. La qualité des maillages est donc relativement médiocre, notamment au niveau de la continuité des patches surfaciques parfois fortement irrégulière.

Les deux autres partenaires engagés dans le sous-projet indexation de

4. `http://shape.cs.princeton.edu/benchmark/`

SEMANTIC-3D ont développé trois méthodes avec lesquelles nous comparons la nôtre :

- DH3DO : une version optimisée de la transformée de Hough 3D (Zaharia et Prêteux 2004) ;
- DH3DO-QV : le descripteur précédent mais avec quantification vectorielle (Zaharia et Prêteux 2004) ;
- ART-3D : une extension d'un descripteur d'images basé moments adopté par MPEG-7 comme descripteur de région 2D (Ricard et al. 2005)

Base du *Princeton Shape Benchmark*

La base du *Princeton Shape Benchmark* (PSB) (Shilane et al. 2004) a été créée en 2004 et est rapidement devenue la collection la plus utilisée au monde pour la comparaison des méthodes de recherche d'objets tridimensionnels. Cette collection est constituée de 1814 modèles 3D collectés à partir de près de 300 sites web différents. Les modèles sont classifiés afin d'être utilisés pour l'évaluation de méthodes d'indexation de données 3D. Pour nos expériences, nous avons utilisé la classification la plus fine comprenant 161 classes.

Les treize méthodes de l'état de l'art avec lesquelles nous avons comparé notre méthode sont : D2 Shape Distribution (Osada et al. 2001) ; Extended Gaussian Image (EGI) (Horn 1984) ; Complex Extended Gaussian Image (CEGI) (Kang et Ikeuchi 1991) ; Q-Gram Statistics Descriptor (2-GR) (Ivanko et Perevalov 2005) ; trois méthodes de type Shape Histogram (SHELLS, SECTORS et SECSHEL) (Ankerst et al. 1999), Voxel (Shilane et al. 2004), Spherical Extent Function (EXT) (Saupe et Vranic 2001), Radialized Spherical Extent Function (REXT) (Vranic 2003), Gaussian Euclidean Distance Transform (GEDT) (Kazhdan et al. 2003), Spherical Harmonic Descriptor (SHD) (Kazhdan et al. 2003), et Light Field Descriptor (LFD) (Chen et al. 2003).

Ces méthodes sont décrites en détails par leurs auteurs dans les publications respectivement citées, et rapidement résumées à la section IV.A de la publication *A Bayesian 3D Search Engine using Adaptive Views Clustering* (Filali Ansary et al. 2007) reproduite en annexe A page 144 de ce mémoire. Dans cette même publication est également expliqué le pré-traitement appliqué aux modèles 3D de la collection pour calculer les descripteurs.

2.2.2 Critères d'évaluation

Afin d'évaluer les performances des méthodes d'indexation, il existe de nombreux indicateurs.

Les partenaires du projet RNRT SEMANTIC-3D ont défini six critères algorithmiques pour comparer les descripteurs de modèles 3D qu'ils ont proposés, ainsi que deux critères de performance. Ces critères ont fait l'objet du livrable L2.4 du projet (Zaharia et al. 2005). Les critères de performance sont : *First Tier* (FT) – proposé par Osada et al. (2001) – et *Score Bull-Eye* (SBE) – retenu comme critère d'évaluation de tous les descripteurs de formes considérés dans MPEG-7.

Le *Princeton Shape Benchmark* (PSB) (Shilane et al. 2004) fournit des outils statistiques pour l'évaluation des méthodes : *Nearest Neighbour* (NN), *First Tier* (FT), *Second Tier* (ST), *E-measure*, et *Discounted Cumulative Gain* (DCG).

Enfin, nous utilisons les courbes rappel-précision qui sont un outil de visualisation des performances globales de tout algorithme de recherche d'information.

Tous ces indicateurs sont décrits en détails dans la section IV.B de la publication *A Bayesian 3D Search Engine using Adaptive Views Clustering* (Filali Ansary et al. 2007) reproduite en annexe A page 144 de ce mémoire.

2.2.3 Expériences et résultats

Les deux bases sur lesquelles nous avons conduit nos expérimentations sont très différentes. Il est donc intéressant de commenter les résultats obtenus sur chacune d'elles.

Sur la base Renault du projet RNRT SEMANTIC-3D

Dans notre comparaison des performances, nous avons également mis en parallèle aux quatre méthodes développées par les partenaires du projet, la méthode *Spherical Harmonic Descriptor* (SHD) (Kazhdan et al. 2003), largement utilisée et comparée dans le domaine de la recherche de modèles 3D.

La figure 2.8 montre un exemple de modèle 3D présenté en requête à notre moteur de recherche, puis les huit premiers résultats par ordre de pertinence.

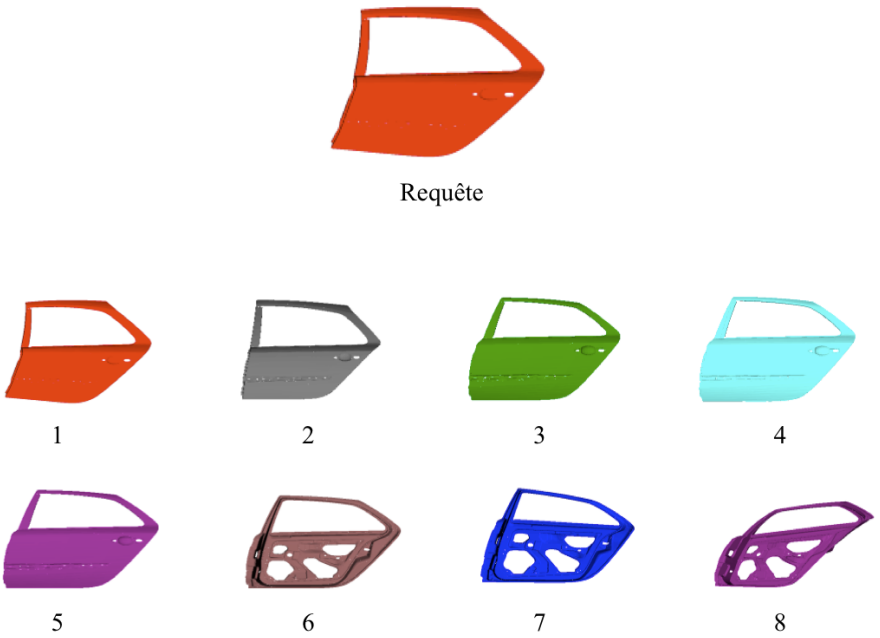


FIGURE 2.8 – *Un objet-requête et les huit premiers résultats par ordre de pertinence.*

Critères	SHD	DH3DO	DH3DO-QV	ART3D	AVC
FT (758)	0.92	0.95	0.93	0.89	0.92
SBE (758)	0.94	0.97	0.96	0.92	0.95
FT (5000)	0.80	0.86	0.83	0.78	0.88
SBE (5000)	0.91	0.94	0.92	0.84	0.94

TABLE 2.1 – *Performances des méthodes sur la collection RNRT SEMANTIC-3D.*

Le tableau 2.1 montre les performances des méthodes selon les critères retenus par les partenaires du projet RNRT SEMANTIC-3D (Zaharia et al. 2005) (*First Tier* et *Score Bull-Eye*). Ces critères ont été calculés sur la collection réduite à 758 modèles répartis en 75 classes, et sur la collection complète de 5000 modèles (constituée de la base réduite et des 4242 autres modèles dans la catégorie "autres").

Remarquons tout d'abord les très bonnes performances de toutes les méthodes proposées dans le cadre du projet RNRT SEMANTIC-3D, ainsi que celle de la méthode SHD. Cela est dû directement à la composition de la collection de modèles 3D : des pièces mécaniques réparties en classes, où chaque classe représente un type de pièce mécanique en différentes versions qui diffèrent très légèrement les unes des autres. Par exemple, les figures 2.3 (page 41) et 2.8 illustrent parfaitement les différentes versions de la porte avant-gauche de la Laguna II, véhicule Renault dont sont extraits tous les modèles de la collection du projet. On peut dire que les variations de formes intra-classe sont très faibles pour cette collection

de modèles 3D. Néanmoins, grâce à l'utilisation des moments de Zernike dans notre approche AVC, la présence ou l'absence de trous dans les objets est bien prise en compte comme l'illustre la figure 2.8. Dans cette figure, la requête est une portière sans trou (hormis le passage de la vitre) et on remarque que les cinq premiers résultats retournés ne présentent aucun trou, comme la requête.

Méthodes	Critères				
	NN	F T	ST	E-Measure	DCG
AVC (probabilités)	99.2%	88.3%	94.7%	52.0%	96.3%
AVC (simple distance)	98.1%	86.8%	93.1%	51.3%	93.9%

TABLE 2.2 – *Performances de la méthode AVC sur la collection RNRT SEMANTIC-3D.*

Le tableau 2.2 montre les performances de notre méthode AVC sur la collection RNRT SEMANTIC-3D selon les critères définis par le *Princeton Shape Benchmark* (Shilane et al. 2004). On remarque aisément que l'utilisation de l'indexation probabiliste – dont le principe est présenté à la section 2.1.4 de ce chapitre – apporte un gain de performance non négligeable de 3 points de pourcentage. La courbe rappel-précsision de la figure 2.9 montre également ce gain très sensible de l'utilisation des probabilités.

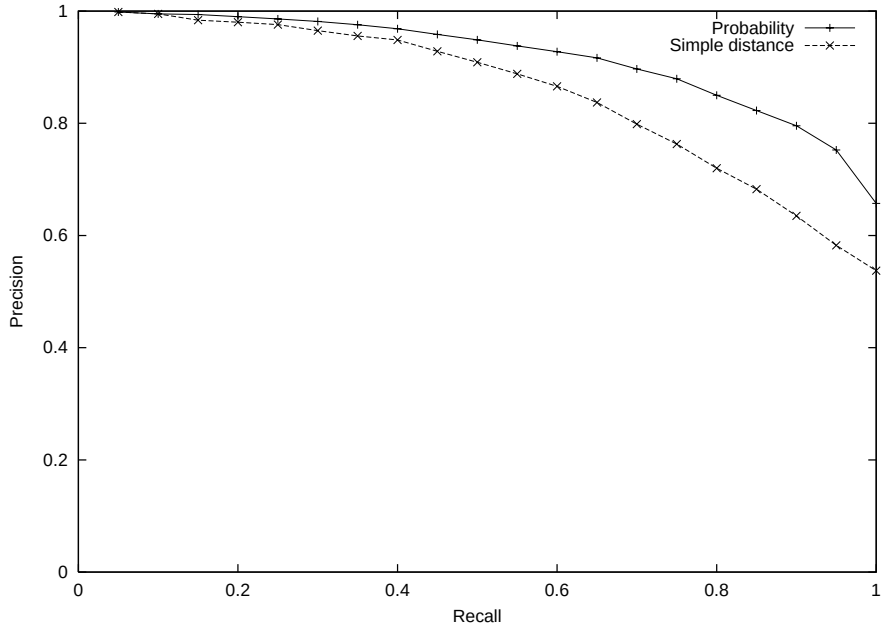


FIGURE 2.9 – *Courbes rappel-précision pour la méthode AVC avec et sans probabilité sur la collection RNRT SEMANTIC-3D.*

Sur la base du *Princeton Shape Benchmark*

Grace aux outils statistiques fournis par le *Princeton Shape Benchmark* (Shilane et al. 2004), nous avons comparé notre méthode AVC avec treize autres méthodes de l'état de l'art.

Méthodes	Critères						
	Taille	NN	FT	ST	E-Meas.	DCG	N-DCG
LFD	4 700	65.7%	38.0%	48.7%	28.0%	64.3%	21.3%
AVC (proba)	**1 113**	**60.6%**	**33.2%**	**44.3%**	**25.5%**	**60.2%**	**13.48%**
REXT	17 416	60.2%	32.7%	43.2%	25.4%	60.1%	13.3%
GEDT	32 776	60.3%	31.3%	40.7%	23.7%	58.4%	10.2%
AVC (dist.)	**1 113**	**58.2%**	**31.1%**	**42.7%**	**25.1%**	**59.9%**	**11,8%**
SHD	2 184	55.6%	30.9%	41.1%	24.1%	58.4%	10.2%
2-GR	512	55.5%	28.7%	39.1%	23.0%	56.3%	—%
EXT	552	54.9%	28.6%	37.9%	21.9%	56.2%	6.0%
SECSHEL	32 776	54.6%	26.7%	35.0%	20.9%	54.5%	2.8%
VOXEL	32 776	54.0%	26.7%	35.3%	20.7%	54.3%	2.4%
SECTORS	552	50.4%	24.9%	33.4%	19.8%	52.9%	-0.3%
CEGI	2 056	42.0%	21.1%	28.7%	17.0%	47.9%	-9.6%
EGI	1 032	37.7%	19.7%	27.7%	16.5%	47.2%	-10.9%
D2	136	31.1%	15.8%	23.5%	13.9%	43.4%	-18.2%
SHELLS	136	22.7%	11.1%	17.3%	10.2%	38.6%	-27.3%

TABLE 2.3 – *Performances des méthodes sur le* Princeton Shape Benchmark.

Le tableau 2.3 montre les performances, selon les critères du benchmark, des méthodes citées à la section 2.2.1 de ce chapitre. La première colonne nommée "taille" donne une indication supplémentaire quant à la taille moyenne d'un descripteur de modèle 3D, critère qui peut avoir son importance en terme de stockage et éventuellement de rapidité de traitement des descripteurs. La figure 2.10 montre les courbes rappel-précision de quelques unes des meilleures méthodes.

On remarque immédiatement que les méthodes utilisant des vues 2D (LFD et AVC) fournissent les meilleurs résultats dans ce benchmark. L'utilisation de vues caractéristiques apporte donc un maximum d'information sur les modèles 3D à indexer.

Si notre méthode AVC probabiliste se classe donc deuxième derrière la méthode Light Field Descriptor (LFD), il convient de préciser que cette dernière utilise un nombre fixe de vues (100 exactement) pour chaque

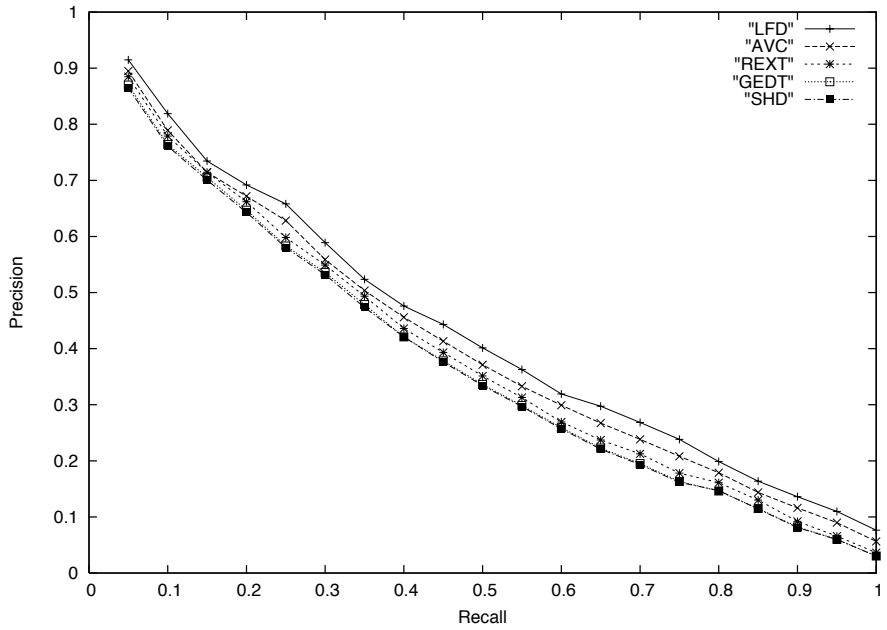


FIGURE 2.10 – *Courbes rappel-précision pour quelques méthodes sur le* Princeton Shape Benchmark.

modèle 3D, là où notre méthode utilise un nombre variable de vues (23 en moyenne) selon la complexité géométrique de l'objet 3D. De plus, la méthode LFD utilise deux descripteurs pour chaque vue (des moments de Zernike et un descripteur de Fourier). Ceci fait que la taille du descripteur global pour un objet 3D est plus importante que celle du descripteur utilisé dans notre méthode.

Enfin, notons à nouveau que l'utilisation de probabilités dans notre méthode AVC donne de meilleurs résultats que sans (ie. avec un simple calcul de distance entre descripteurs), comme nous l'avons déjà remarqué dans les expériences précédentes sur la collection du projet RNRT SEMANTIC-3D.

D'autres expériences ont été faites concernant notamment la robustesse au bruit et à la décimation de facettes dans le maillages. Elles sont décrites en détails dans la section IV.C de la publication *A Bayesian 3D Search Engine using Adaptive Views Clustering* (Filali Ansary et al. 2007) reproduite en annexe A page 144 de ce mémoire.

Pour conclure les expériences, nous pouvons affirmer que notre méthode AVC probabiliste offre un très bon compromis entre la qualité (performance de la recherche) et le coût (stockage) pour le *Princeton Shape Benchmark*.

2.2.4 Quelques résultats utilisant des photographies

Jusqu'à présent dans ce chapitre, nous avons utilisé notre approche AVC avec des requêtes purement 3D. La méthode AVC utilisant des vues 2D pour caractériser les objets 3D, une requête 3D est caractérisée par l'ensemble de ses vues. Rien n'empêche donc de présenter une requête sous la forme d'une seule et unique vue 2D (dessin, croquis, photographie, etc.), voire de plusieurs vues 2D conjointes.

Pour évaluer les performances de notre approche en utilisant des vues 2D pour requêtes, nous avons sélectionné cinquante images différentes sur l'Internet. Ces images correspondent à dix classes du *Princeton Shape Benchmark*. Après quelques transformations (passage en niveaux de gris, extraction des contours par le filtre de Canny (1986), binarisation, normalisation et centrage) pour permettre et faciliter le calcul des moments de Zernike, les images de chaque classe sont utilisées comme requêtes, seules ou conjointement avec d'autres images de la même classe.

Les figures 2.11 à 2.13 montrent les résultats renvoyés par notre moteur de recherche en utilisant une, deux, puis trois photos de bicyclettes en requête. Les résultats sont classés par ordre de pertinence de haut en bas et de gauche à droite. On remarque que dès l'utilisation d'une seule image-requête, un modèle 3D de bicyclette apparaît déjà en deuxième position (figure 2.11). Deux images-requêtes donnent de meilleurs résultats avec quatre modèles 3D de bicyclettes dans les douze premiers, dont trois aux premières places (figure 2.12). Enfin, trois images-requêtes donnent d'encore meilleurs résultats avec la présence de cinq modèles 3D pertinents dont quatre aux premières places (figure 2.13).

Les résultats ne sont donc pas "parfaits" mais offrent une première aide performante dans la recherche ou la navigation à travers une collection de modèles 3D, si on permet à l'utilisateur d'utiliser les résultats fournis par le moteur de recherche comme requête 3D pour une nouvelle recherche. C'est ce que permet notre moteur de recherche sur le Web décrit dans la section suivante.

Ces résultats, ainsi que des expériences plus détaillées – comme des courbes rappel-précision par image et par classe d'images – ont fait l'objet d'une publication dans la conférence *IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT)* en 2006 (Filali Ansary et al. 2006).

FIGURE 2.11 – *Requête avec une seule image et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite).*

FIGURE 2.12 – *Requête avec deux images et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite).*

FIGURE 2.13 – *Requête avec trois images et résultats classés par ordre de pertinence du premier au douzième (de haut en bas et de gauche à droite).*

2.3 Moteur de recherche 3D sur le Web

Pour permettre à tout à chacun de tester notre méthode, nous avons développé un moteur de recherche en ligne accessible sur le Web à l'adresse `http://www-rech.telecom-lille1.eu/3dretrieval/`. Il est accessible à tous les navigateurs web modernes – c'est-à-dire compatibles Java notamment – que ce soit sur ordinateurs classiques, assistants numériques (PDA), ou *smartphones* (iPhones, Android Phones). Dans ces deux derniers cas, une interface adaptée à la taille de ce type d'écran est utilisée.

Le moteur de recherche en ligne implante la méthode AVC probabiliste et permet des tests sur 1814 modèles 3D de la collection du *Princeton Shape Benchmark*[5]. Les résultats sont présentés sous la forme d'imagettes et des liens vers les modèles 3D originaux sont disponibles.

Comme le montrent les figures 2.16 et 2.17, l'interface est divisée en deux parties : à gauche, l'interface de requête ; à droite l'affichage des résultats. La partie de gauche offre trois modalités d'expression de la requête : requête-image (un maximum de trois images est utilisable simultanément), parcours de la collection par le nom des modèles, et dessin à la souris. Cette dernière modalité est en cours de développement.

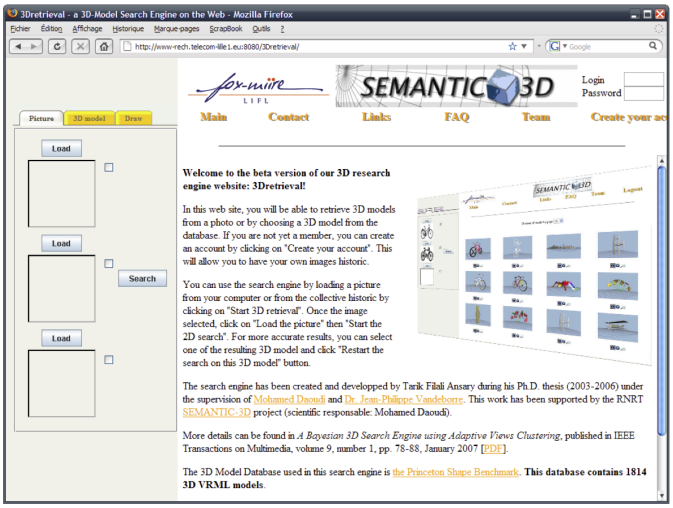


Figure 2.14 – *Interface web du moteur de recherche en ligne avant toute opération.*

Les figures 2.14 à 2.17 illustrent un exemple d'utilisation du moteur de recherche en ligne. La figure 2.14 montre l'interface avant toute opération : l'interface de requête à gauche et un texte d'accueil à droite. La partie de

5. Pour des raisons de droits, la collection du projet RNRT SEMANTIC-3D, propriété du constructeur automobile Renault, ne peut pas être mise en ligne sur un site web ouvert au public

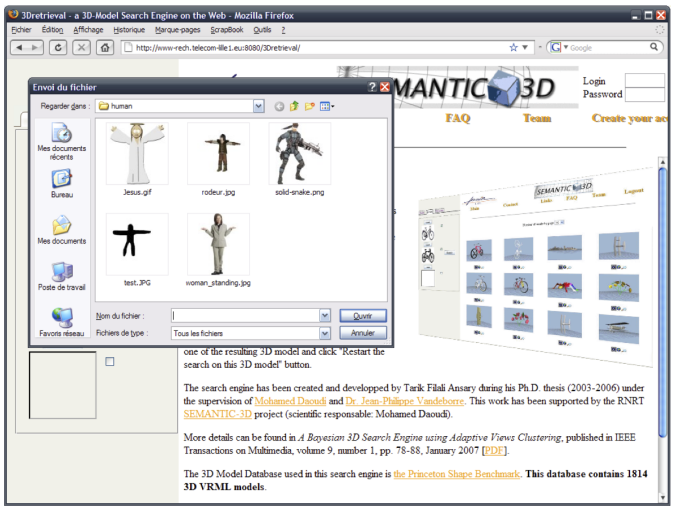


FIGURE 2.15 – *Envoi d'une image-requête par l'utilisateur.*

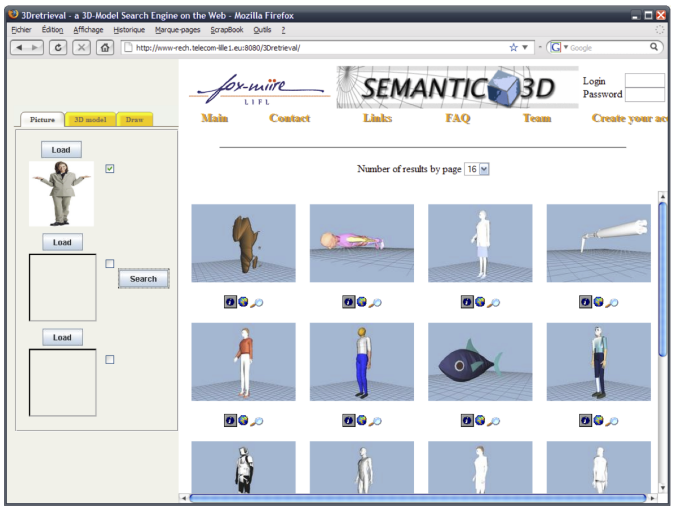


FIGURE 2.16 – *Résultats de la recherche à partir de l'image-requête unique.*

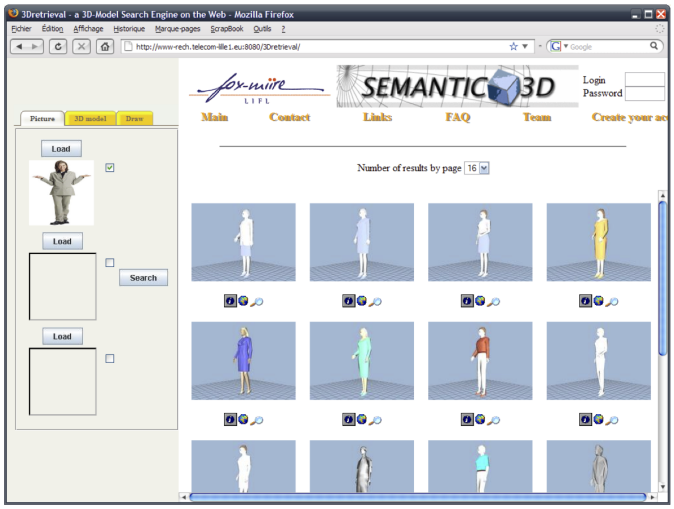


FIGURE 2.17 – *Résultats de la recherche à partir d'un modèle 3D, résultat de la requête initiale (troisième modèle 3D dans les résultats de la figure 2.16).*

droite sera remplacée par les résultats par la suite. Tout d'abord, l'utilisateur transfert une image-requête depuis son ordinateur (figure 2.15). Puis il lance une recherche à partir de cette image-requête unique en cliquant sur le bouton *search*. Au bout de quatre à cinq secondes, les résultats envoyés par le moteur de recherche apparaissent dans la partie de droite (figure 2.16). L'utilisateur sélectionne alors l'un de ces résultats – en cliquant sur la troisième imagette par exemple – pour relancer une requête prenant cette fois en compte un modèle 3D entier. Après à nouveau quelques secondes d'attente, des résultats plus pertinents sont enfin affichés dans la partie droite (figure 2.17).

Concernant le temps d'attente des résultats, il faut noter qu'il correspond à l'ensemble des traitements effectués depuis l'envoi de l'image-requête jusqu'à l'affichage des résultats. Cette durée comprend donc l'envoi de l'image-requête du navigateur web vers le serveur, les calculs sur le serveur – c'est-à-dire le pré-traitement de l'image-requête, l'extraction du descripteur de celle-ci, le calcul des probabilités utilisant le descripteur-requête et les descripteurs stockés sur le serveur (calculés quant à eux pendant la phase *offline* décrite à la section 2.1.1), la sortie des résultats par ordre de pertinence, et l'envoi des imagettes de résultats vers le navigateur web –, et enfin l'affichage de ces imagettes dans le navigateur. Le temps de calcul brut des comparaisons entre la requête et les descripteurs stockés dans la base du serveur représente moins de la moitié de ce temps d'attente. Le temps d'envoi/réception des données par le réseau dépend bien sûr de la connexion à Internet.

Pour finir cette partie, signalons qu'une application mobile dédiée (pour appareils fonctionnant sous Android) est en cours de développement. Elle permettra d'accéder plus facilement au moteur de recherche en ligne sur *smartphone* et autre ordinateurs-tablettes.

2.4 Conclusion du chapitre

Nous avons présenté dans ce chapitre les travaux de recherche menés en indexation de modèles 3D avec l'encadrement de la thèse de Tarik Filali Ansary. L'objectif de ces travaux était de proposer une méthode d'indexation de modèles 3D permettant la recherche à l'aide d'une ou plusieurs photographies de l'objet recherché. L'approche de la *recherche par l'exemple* est unanimement reconnue dans ce domaine et elle nous est bien évidemment apparue incontournable dès le début de nos réflexions.

Le premier apport de nos travaux se trouve dans l'idée qu'un objet tridimensionnel se représente intuitivement et facilement par un certain nombre de vues que nous avons appelées *vues caractéristiques*. Ces vues ne sont d'ailleurs pas nécessairement en nombre défini et fixé à l'avance, mais bien en nombre variable en fonction de la complexité géométrique de l'objet concerné : plus un objet est complexe, plus il nécessitera de vues pour le caractériser. Grace à l'utilisation de vues, la comparaison de deux objets 3D est simple et intuitive : deux objets 3D sont similaires si les vues qui les caractérisent le sont aussi.

Le second apport de nos travaux réside dans la façon d'utiliser ces vues pour indexer et retrouver les modèles 3D. Nous sommes en effet parti du constat que toutes les vues d'un objet 3D n'ont pas la même importance pour le caractériser. Certaines vues portent en effet plus d'information en elles que d'autres, dans le sens où elles sont plus susceptibles d'être représentées dans une requête. De plus, des objets simples (comme un cube) sont souvent à la base d'objets plus complexes. Ces objets auront par conséquent des vues caractéristiques communes. Nous avons donc pris ces considérations en compte et intégré l'utilisation de probabilités dans notre méthode d'indexation.

Ces travaux répondent – en partie bien sûr, car d'autres pistes restent encore à explorer [6] – à la première des deux grandes questions soulevées dans l'introduction de ce mémoire : « Comment retrouver un modèle 3D dans une collection ? »

Bien que les vues caractéristiques, comme l'indique leur nom, représentent l'objet tridimensionnel avec le maximum d'information visuelle dans le but de l'indexer puis, par la suite, le retrouver dans une collection, ces vues n'apportent aucune connaissance sur la structure de l'objet lui-même ou, pour utiliser d'autres termes, sur les différentes parties qui le composent. C'est ainsi que nous nous sommes intéressés à la seconde question posée dans l'introduction de ce document : « Comment retrouver des informations de structure d'un modèle 3D maillé sans connaissance *a priori* sur celui-ci ? » Cette question est le sujet du chapitre suivant.

6. Nous avons à ce propos exploré d'autres pistes pour la sélection automatique des vues caractéristiques d'un objet 3D : l'utilisation des enveloppes visuelles pour déterminer le nombre et la position des vues (Theetten et al. 2005), et les positions d'équilibre mécanique pour trouver les vues les plus susceptibles d'être utilisées en tant qu'images-requêtes (Theetten et al. 2008).

3 Analyse structurelle de maillages 3D

Selon les applications et les besoins en terme d'affichage ou de traitement, les modèles d'objets tridimensionnels sont représentés de manières et dans des formats différents : nuage de points, représentation surfaciques, représentation volumique, formats propriétaires et spécifiques aux applications, etc. Mais force est de constater – sur l'Internet, dans les bases multimédias, les benchmarks, etc. – que c'est certainement la représentation surfacique (*boundary representation* ou *B-rep*) qui est la plus répandue. La représentation surfacique elle-même peut être subdivisée en plusieurs catégories. Parmi celles-ci, le maillage de polygones est de loin la représentation de surface la plus répandue. En effet, cette représentation de la surface d'un objet par un ensemble de facettes polygonales assemblées est parfaitement adaptée à l'affichage de l'objet par la carte graphique spécialisée d'un ordinateur.

Dans le chapitre précédent, nous nous intéressions à l'indexation de modèles 3D par le biais de vues caractéristiques. Pour obtenir une vue, connaissant la position et la direction du point de vue, il suffit simplement d'effectuer un rendu, c'est-à-dire une projection de l'objet sur le plan image. Dans notre cas, ce rendu peut être très simple – objet noir sur fond blanc – car le descripteur utilisé ne nécessite ni couleurs ni textures – informations que, de toutes les façons, nous ne possédons pas –, ni nécessairement de détails élaborés. En d'autres termes, quel que soit la représentation ou le format utilisé pour définir les modèles 3D, à partir du moment où le calcul d'un rendu simple est possible, notre méthode d'indexation fonctionne.

Puisque la modélisation par un maillage de polygones est la représentation 3D la plus répandue, nous avons plus particulièrement porté notre

attention sur celle-ci pour répondre à la deuxième question soulevée dans l'introduction de ce mémoire : « Comment retrouver des informations de structure d'un modèle 3D maillé sans connaissance *a priori* sur celui-ci ? » Pour cela, nous nous sommes intéressés au passage d'une représentation discrète (le maillage de polygones) à une représentation plus expressive et intrinsèque de la forme tridimensionnelle. Cette nouvelle représentation nous a ensuite permis de proposer des solutions à trois applications liées au traitement de formes 3D : la manipulation interactive de formes, l'indexation 3D partielle, et la segmentation de formes 3D.

Dans ce chapitre, nous présentons les recherches que nous avons menées avec l'encadrement de la thèse de Julien Tierny soutenue le 2 octobre 2008. Le chapitre se décompose de la façon suivante. Une première partie présente les principaux types de représentation d'objets 3D, en mettant l'accent sur le maillage de polygones. Nous présentons ensuite le coeur du travail concernant les squelettes topologiques améliorés. Enfin, des solutions aux trois applications liées au traitement de formes 3D, citées précédemment, sont présentées.

3.1 Représentations de formes 3D

Il existe de nombreuses manières de représenter la forme des objets 3D. Celles-ci peuvent être classées grossièrement en deux catégories : les représentations volumiques et les représentation surfaciques.

3.1.1 Représentations volumiques

Dans le cas des représentations volumiques, la forme d'un objet 3D est définie en fonction du volume de celui-ci (*ie.* la place que l'objet 3D occupe dans l'espace). Alors que dans le cas des représentations surfaciques, seule la surface externe est prise en compte dans la représentation de la forme.

La grille de *voxels* est certainement l'exemple le plus connu en représentation volumique. Le terme *voxel* est une analogie avec le terme *pixel*. Là où *pixel* (*picture element*) désigne un élément de base de la grille d'une image 2D *bitmap*, le terme *voxel* (*volume element*) désigne un volume élémentaire d'une grille qui subdivise l'espace euclidien 3D. L'énumération des *voxels* occupés par l'objet 3D défini alors sa forme volumique. Cette représentation est particulièrement adaptée aux applications où ce qui est "à l'intérieur" d'un objet est aussi important – sinon plus – que sa surface

extérieure. Ceci est particulièrement vrai pour les applications médicales pour lesquelles il est important de modéliser les parties internes des organes pour en obtenir des coupes par la suite.

3.1.2 Représentations surfaciques

Les représentations surfaciques prennent uniquement en compte la surface extérieure des objets. Ces représentations nécessitent donc moins de mémoire que leur équivalent volumique (pour un niveau de détail équivalent). De plus, la forme d'un objet 3D est principalement donnée par sa surface extérieure. C'est ce à quoi s'intéresse la plupart des applications des domaines de la synthèse d'images et de la vision par ordinateur.

Les représentations surfaciques sont elles-mêmes divisées en deux catégories : les surfaces implicites et les surfaces explicites.

Surfaces implicites. Une surface implicite est la surface de niveau d'une fonction différentiable f définie sur un ouvert de $\mathbb{R}^3$. Soit une fonction $f : \mathbb{R}^3 \to \mathbb{R}$, la surface implicite S associées à f est l'ensemble des points P de coordonnées (x, y, z) qui vérifient l'équation $f(x, y, z) = 0 \Leftrightarrow S = f^{-1}(0)$.

Ces surfaces, par leurs définitions mathématiques simples et leurs bonnes propriétés différentielles, sont très utilisées pour la modélisation d'objets d'aspect lisse, comme dans les applications de Conception Assistée par Ordinateur (CAO). Cependant, le rendu efficace de telles surfaces passe généralement par une conversion vers une représentation explicite, et notamment vers une représentation maillée. L'ouvrage dirigé par Bloomenthal (1997) est une excellente référence sur les surfaces implicites.

Surfaces explicites. Au contraire des surfaces implicites, une surface explicite est entièrement définie par des fonctions mathématiques. Les surfaces explicites les plus connues sont les surfaces paramétriques – comme les surfaces de Bézier, les NURBS, etc. – et les surfaces maillées.

De la même façon que pour les surfaces implicites, les surfaces paramétriques nécessitent un passage vers une représentation maillée pour être affichées efficacement.

3.1.3 Un peu plus sur les maillages de polygones...

Un maillage est l'approximation linéaire par morceaux d'une surface. La représentation explicite d'un maillage est donnée par la définition d'un échantillon de points P de coordonnées (x, y, z) appartenant à la surface. Ces points sont connectés entre eux pour former une structure de graphe. La surface est donc localement approximée par des polygones. Pour assurer l'aspect plan des polygones, ces derniers sont généralement des triangles. On parle alors de maillage triangulaire ou de triangulation d'une surface.

Un maillage M représentant une surface S est donc défini par :

$$M = \{V, E, F\}$$

avec $V = \{v_1, v_2, \ldots, v_{N_V}\}$, $E = \{e_1, e_2, \ldots, e_{N_E}\}$, et $F = \{f_1, f_2, \ldots, f_{N_F}\}$.

V est l'ensemble des points (*vertices*) du maillage – un point du maillage correspond à un point de l'échantillon –, E est l'ensemble des arêtes (*edges*) de la structure de graphe, et F est l'ensemble des facettes (*faces*) correspondant aux polygones du maillage.

Pour stocker un modèle 3D défini par une telle structure de données, il existe de très nombreux formats dédiés comme le *3D Object File Format* (OFF), le format *Stanford PoLYgon* (PLY), le format *GNU Triangulated Surface* (GTS), etc. D'autres formats plus généralistes de définition de scènes tridimensionnelles, comme le *Virtual Reality Modeling Language* (VRML) et son successeur le format X3D [1], offrent la possibilité de définir des objets 3D maillés. Tous ces formats stockent une surface maillée à peu près de la même façon : une liste non organisée des coordonnées des points – l'ordre dans la liste définit la numérotation utilisée par la liste suivante –, suivie d'une liste des polygones donnée par une suite de numéros correspondants aux points de la première liste. La définition d'un polygone se termine par un marqueur spécifique (-1 par exemple car cette valeur ne représente pas un numéro de point dans la première liste), ce qui permet de définir des polygones ayant un nombre d'arêtes arbitraire. Généralement, après lecture du fichier, quelque soit son format, les données sont stockées en mémoire dans une structure optimisant l'accès aux informations : structures orientées faces ou arêtes, dirigées ou non, etc. Le chapitre 2 de l'ouvrage *Polygon Mesh Processing* de Botsch et al. (2010) résume parfaitement les différentes structures de représentation des surfaces maillées en mémoire.

1. X3D est une norme du consortium Web3D – `http://www.web3d.org/x3d/`

Un maillage étant une approximation d'une surface, il existe une infinité de maillages pour représenter une surface donnée. La figure 3.1 montre l'objet 3D *Bimba Con Nastrino* du AIM@SHAPE Shape Repository [2] en trois niveaux de détails – ou résolutions – différents. Notons aussi que, dans cet exemple, les points sont relativement bien répartis sur la surface et que les polygones sont de tailles comparables. Il est tout à fait possible de trouver des zones plus ou moins denses en fonction des détails à représenter dans lesdites zones. La résolution d'un maillage 3D n'est donc pas obligatoirement uniforme.

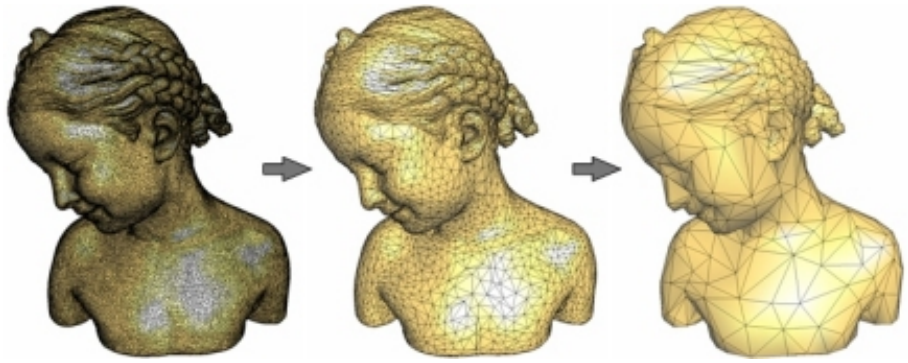


FIGURE 3.1 – Bimba Con Nastrino *(AIM@SHAPE Shape Repository) en différentes résolutions de maillages.*

Enfin, une propriété importante que doit respecter un maillage pour être utilisable – affichage, traitement, etc. – dans de bonnes conditions est d'être *manifold* (ou être une *2-variété* en français). Un maillage est dit *manifold* si chacun de ses sommets est homéomorphe à un disque, ou un demi-disque pour les sommets de bord. Visuellement par exemple, cela signifie qu'une arête ne peut appartenir qu'à une (dans le cas d'une arête de bord) ou deux facettes du maillage.

3.2 Squelette topologique avancé

Nous exposons dans cette partie du chapitre la méthode que nous avons mise au point pour extraire un squelette topologique avancé à partir d'un maillage de polygones représentant une surface.

Nous commençons par faire un rapide résumé des différentes étapes pour aboutir à un squelette topologique en partant d'un maillage de polygones. Puis nous définissons la notion de graphe de Reeb avant de détailler la fonction d'application que nous utilisons pour construire celui-

2. `http://shapes.aimatshape.net/`

ci. Le calcul des points caractéristiques, point de départ de l'algorithme, est détaillé, suivi de la définition des lignes de niveaux discrètes et des contours discrets, pour aboutir à la construction du graphe de Reeb. Nous présentons ensuite une amélioration du graphe de Reeb classique grâce à l'analyse géométrique des contours discrets. Enfin l'obtention du squelette topologique avancé d'un maillage de polygones est expliquée.

3.2.1 Résumé de la méthode

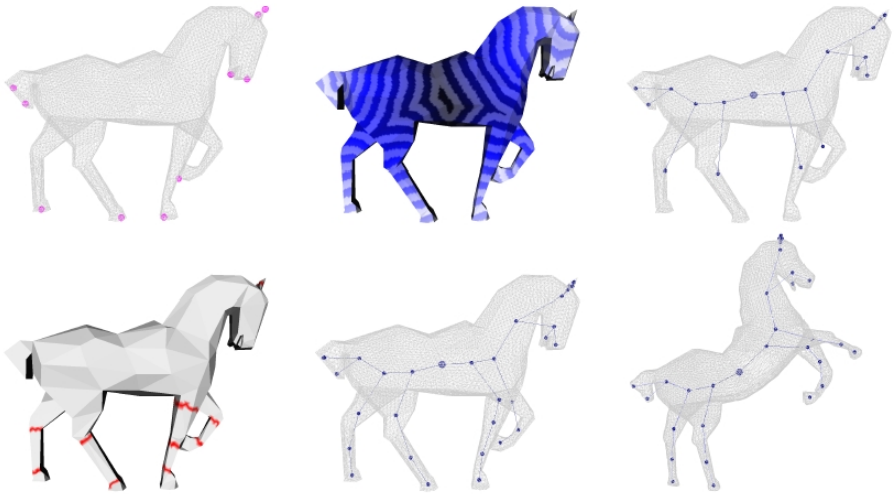


FIGURE 3.2 – *Principales étapes de notre méthode (de gauche à droite et de haut en bas).*

La figure 3.2 donne une vue d'ensemble des différentes étapes de la construction d'un squelette topologique avancé. La première étape consiste à extraire les points caractéristiques du maillage. Puis, pour chaque sommet du maillage, une fonction d'application est calculée. Connaissant la valeur de la fonction d'application en chaque point du maillage, il devient possible de calculer un graphe de Reeb. Pour améliorer les détails rendus par ce graphe, des constrictions sont calculés pour aboutir à un graphe topologique amélioré. Grâce à ce graphe, des applications – comme la déformation de maillages – sont possibles.

Une vidéo résumant ces différentes étapes est visible à l'URL suivante : `http://www.telecom-lille1.eu/people/vandeborre/hdr/chap2resume.avi`.

3.2.2 Graphe de Reeb

Un graphe de Reeb (Reeb 1946) est une structure topologique définie de la manière suivante.

Soit f une fonction réelle définie sur une variété compacte V $f : V \rightarrow \mathbb{R}$. Le graphe de Reeb de f est l'espace quotient de f dans $V \times \mathbb{R}$, par la relation d'équivalence $(p_1, f(p_1)) \sim (p_2, f(p_2))$, vérifiée si et seulement si :

$$\begin{cases} f(p_1) = f(p_2) \\ p_1 \text{ et } p_2 \text{ appartiennent à la même composante} \\ \qquad \text{connexe de } f^{-1}(f(p_1)) \end{cases}$$

Concrètement, un graphe de Reeb est composé de nœuds représentant les points critiques de f, autrement dit les points de la variété V où les dérivées partielles de f s'annulent. Les arêtes du graphe représentent les composantes connexes de V reliant les points critiques de f.

La figure 3.3 présente un graphe de Reeb calculé sur un bitore selon la fonction de hauteur [3]. Les points critiques de cette fonction sont marqués en rouge pour les minima, en vert pour les maxima, et en noir pour les points selles.

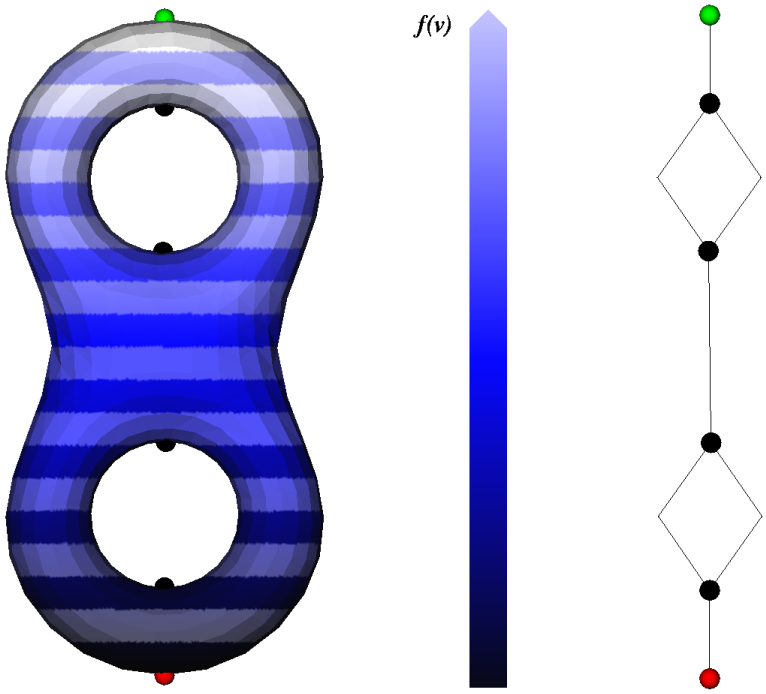


FIGURE 3.3 – *Evolution des lignes de niveau de la fonction hauteur sur un bitore, ses points caractéristiques, et son graphe de Reeb.*

Dans le cas des surfaces triangulées, les algorithmes traditionnels de construction de graphes de Reeb (Tarasov et Vyalyi 1998, Cole-McLaughlin

3. Comme nous le verrons plus loin, la fonction de hauteur n'est pas la fonction d'application idéale pour des modèles 3D génériques. Néanmoins, sa simplicité de représentation convient parfaitement à l'illustration des principes et méthodes présentés dans ce chapitre.

et al. 2003) identifient dans un premier temps les sommets correspondant aux points critiques de f, puis construisent le graphe en analysant leurs relations de connexité.

Cependant, ces méthodes présupposent que l'ensemble des points critiques identifiés est significatif, alors qu'en pratique, cette hypothèse peut conduire à des graphes comportant un grand nombre de nœuds et d'arêtes (Ni et al. 2004), encodant des détails insignifiants et n'offrant donc pas une description globale de haut niveau. Pour fournir une description structurelle globale d'un objet 3D, il est donc indispensable d'apporter une solution à ce problème de sélection des points critiques. Plusieurs algorithmes pratiques ont été proposés à cette fin (Attene et al. 2003, Peer Timo et al. 2004), mais tous sont conditionnés par un paramètre d'entrée.

Dans notre approche, nous proposons un algorithme unifié de construction et de simplification de graphe, ne prenant aucun paramètre d'entrée.

3.2.3 Quelle fonction d'application ?

La fonction d'application f est parfois donnée par le contexte applicatif. C'est le cas en modélisation de terrains par exemple où la fonction qui semble la plus significative est l'élévation. Elle met en effet en valeur les pics et les vallées du terrain. Lorsqu'il s'agit de construire des squelettes topologiques (Biasotti et al. 2003) ou des graphes de Reeb de haut niveau, il est nécessaire de définir une fonction d'application f qui, d'une part, présente des propriétés d'invariance et qui, d'autre part, mette en valeur les parties visuellement intéressantes de l'objet. Dans ce contexte, Lazarus et Verroust (1999) ont proposé une fonction définie en tout sommet de la triangulation par sa distance géodésique – *ie.* la longueur du plus court chemin entre deux sommets d'une triangulation en restant sur le maillage – à un sommet source. Le choix de ce point caractéristique est crucial et peut poser problème dans des applications où la stabilité est une propriété fondamentale. Pour répondre à ce problème, dans le cadre de l'indexation 3D, où cette stabilité a de l'importante, Hilaga et al. (2001) proposent d'intégrer cette fonction sur tout le maillage, au dépend toutefois d'un coût de calcul relativement élevé.

Dans notre approche, pour mettre en valeur la structure globale de l'objet, nous utilisons des distances géodésiques dont les points sources sont les sommets caractéristiques du maillage. Nous proposons en ef-

fet une fonction d'application dont la valeur en chaque sommet v d'un maillage M est la distance géodésique normée du point v concerné au plus proche sommet caractéristique du maillage M.

L'utilisation de la distance géodésique assure une invariance de la fonction d'application aux translations et aux rotations, et garantit également une certaine robustesse aux changements de poses du modèle 3D. De plus, l'invariance au changement d'échelle (homothétie uniforme) est donnée par la normalisation de la distance.

3.2.4 Sommets caractéristiques

Notre fonction d'application étant définie par rapport à des sommets caractéristiques du maillage, leur détermination est une étape cruciale pour la suite de la méthode.

Même si la notion de sommets caractéristiques d'un maillage a été introduite par de nombreux auteurs (Mortara et Patanè 2002, Katz et al. 2005, Liu et Zhang 2007), il reste difficile d'en donner une définition formelle.

Dans notre approche, **les sommets caractéristiques d'un maillage 3D sont les sommets situés aux extrémités des composantes proéminentes de l'objet représenté**.

Pour calculer cet ensemble de sommets caractéristiques, il faut tout d'abord commencer par calculer les deux points du maillages v_{s_1} et v_{s_2} les plus éloignés – au sens géodésique – l'un de l'autre. Deux fonctions sont ensuite définies sur chaque sommet v du maillage M de la manière suivante :

$$f_1(v) = \delta(v, v_{s_1})$$

$$f_2(v) = \delta(v, v_{s_2})$$

où $\delta(v_1, v_2)$ désigne la distance géodésique entre les points v_1 et v_2 du maillage M.

En utilisant la classification des points critiques proposée par Cole-McLaughlin et al. (2003), nous définissons un minimum local (respectivement un maximum local) comme un sommet du maillage dont tous les voisins directs ont une valeur de fonction supérieure (respectivement inférieure). Soient E_1 et E_2 les ensembles des extrema locaux (minima et maxima) des fonctions f_1 et f_2 respectivement, l'ensemble des sommets

caractéristiques F du maillage M est l'intersection de ces deux ensembles E_1 et E_2 :

$$F = E_1 \cap E_2$$

Cependant, en pratique, les extrema locaux de f_1 et f_2 qui correspondent à des sommets caractéristiques, n'apparaissent pas exactement sur les mêmes points du maillage, mais dans un voisinage géodésique que nous avons défini expérimentalement.

Tous les détails concernant l'extraction des sommets caractéristiques d'un maillage 3D sont donnés dans la section 4 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document.

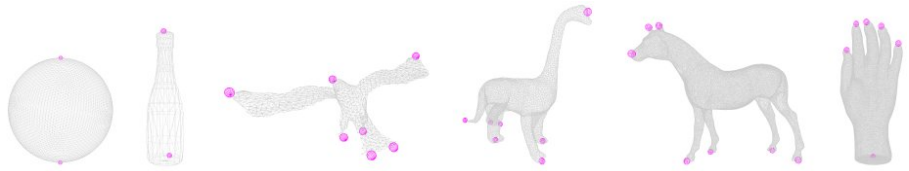


FIGURE 3.4 – *Sommets caractéristiques de quelques maillages 3D.*

On peut facilement se représenter visuellement les sommets caractéristiques. Ils donnent une vue globale de la structure du modèle 3D. La figure 3.4 montre les sommets caractéristiques de plusieurs objets tridimensionnels maillés. On remarque que la méthode donne d'excellents résultats avec des objets relativement complexes – une main, un cheval, etc. – mais également avec des objets très simples comme une sphère ou une bouteille, où seuls deux sommets caractéristiques sont trouvés puisque ces objets ne présentent pas de réelles composantes proéminentes.

3.2.5 Notre fonction d'application

Comme nous l'avons précédemment souligné, nous souhaitons mettre en valeur la structure globale du modèle 3D. Nous utilisons donc les sommets caractéristiques et la distance géodésique des points du maillage vers ces sommets. Notre fonction d'application f_a est par conséquent définie comme suit :

$$f_a(v) = 1 - \hat{\delta}(v, v_p)$$

avec v_p le sommet caractéristique le plus proche de v :

$$v_p \in F \ / \ \hat{\delta}(v, v_p) = min_{v_{f_i} \in F} \ \delta(v, v_{f_i})$$

Tous les détails concernant l'extraction de la fonction d'application, et notamment sa normalisation, sont donnés dans la section 5 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document.

FIGURE 3.5 – *Évolution des lignes de niveau de f_a et ses points critiques (en rouge et en noir). Les points verts sont les sommets caractéristiques.*

La figure 3.5 montre des exemples de calcul de f_a sur deux modèles ainsi que leurs sommets caractéristiques (en vert) et les points critiques de f_a identifiés selon la classification proposée par Cole-McLaughlin et al. (2003). Le modèle *main* possède 6 sommets caractéristiques et 94 points critiques. Le modèle *cheval* possède quant à lui 7 sommets caractéristiques et 92 points critiques. En utilisant les algorithmes traditionnels de construction de graphes de Reeb (Tarasov et Vyalyi 1998, Cole-McLaughlin et al. 2003), ce nombre important de points critiques génère des graphes complexes comportant un grand nombre de nœuds (en fait, autant que de points critiques). Ces graphes montrent alors des détails de moindre importance et cachent la description globale de haut niveau que nous souhaitons atteindre.

Nous apportons une solution à ce problème avec la notion de *ligne de niveau discrète*.

3.2.6 Lignes de niveau discrètes et contours discrets

La définition de lignes de niveau d'une fonction réelle f définie sur une maillage M n'est pas triviale.

En effet, cette définition dans le cas continu est relativement aisée : deux points p_1 et p_2 appartiennent à la même ligne de niveau $f^{-1}(f(p_1))$

si $f(p_2) - f(p_1) = 0$. De plus, p_1 et p_2 appartiennent au même contour si ils appartiennent à la même composante connexe de $f^{-1}(f(p_1))$.

Par contre, dans le cas discret, pour un point donné $v \in M$, selon l'échantillonnage de M, $f^{-1}(f(v))$ est souvent réduite au point v lui même. Au vu de la définition du graphe de Reeb (voir paragraphe 3.2.2 page 66 de ce chapitre), un graphe de Reeb correct ne pourrait pas être calculé avec cette définition des lignes de niveau continues appliquée directement au discret. Les conditions strictes de la relation d'équivalence seraient en effet rarement vérifiées.

Pour préserver les propriétés topologiques des lignes de niveau dans le cas discret, nous avons défini la *ligne de niveau discrète* $\Gamma(v)$ associée au point v par une courbe portée par les arêtes de M, approximant par valeur supérieure la ligne de niveau continue $f^{-1}(f(v))$.

Nous appelons également *contour discret* chaque sous-ensemble connexe de $\Gamma(v)$. En particulier, nous définissons le contour discret $\gamma(v)$ associé au point v comme le sous-ensemble connexe de $\Gamma(v)$ contenant v. Bien évidemment, plus l'échantillonnage de M sera important, plus les lignes de niveau discrètes tendront vers les lignes de niveau continues.

Les détails concernant les lignes de niveau discrètes et les contours discrets sont donnés dans la section 6 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document.

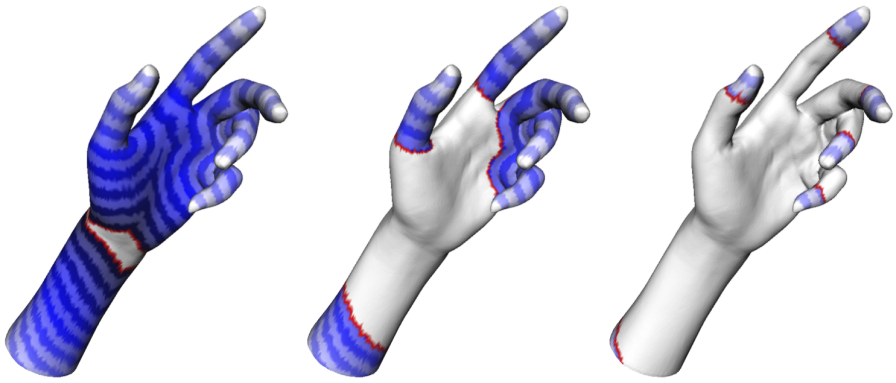


FIGURE 3.6 – *Exemples de lignes de niveau discrètes sur le modèle* main *composé de 25 000 points :* $\Gamma(v_{600})$ *présentant 2 contours,* $\Gamma(v_{10\,000})$ *présentant 4 contours, et* $\Gamma(v_{20\,000})$ *présentant 6 contours.*

Un algorithme itératif de type remonté de gradient est utilisé pour calculer les lignes de niveau discrètes. La figure 3.6 illustre plusieurs exemples de lignes de niveau discrètes sur le modèle 3D *main* constitué

de 25 000 sommets. Explorer récursivement $\Gamma(v)$ permet d'identifier chacun de ses sous-ensembles connexes, et en particulier $\gamma(v)$.

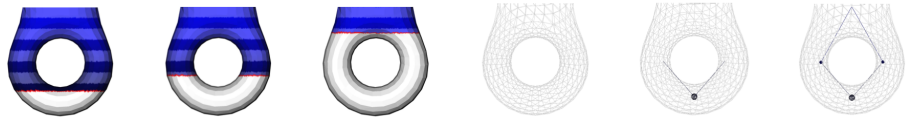
3.2.7 Construction du graphe de Reeb de haut niveau

La construction du graphe de Reeb s'appuie sur les lignes de niveau discrètes $\Gamma(v)$ et sur les contours discrets $\gamma(v)$ décrits dans le paragraphe précédent.

L'idée générale de l'algorithme de construction du graphe de Reeb est d'étudier l'évolution des sous-ensembles de $\Gamma(v)$ et d'identifier chacune des composantes connexes de M traversée par $\Gamma(v)$. Dans notre méthode, la construction du graphe de Reeb se fait pendant l'algorithme de construction des lignes de niveau discrètes. Au fur et à mesure de la construction des lignes de niveau discrètes, on relève les variations topologiques parmi les trois suivantes :

- la bifurcation, quand deux composantes se créent à partir d'une seule à l'itération précédente ;
- la jonction, quand deux composantes se rejoignent en une seule à l'itération suivante ;
- la terminaison, quand une composante disparaît d'une itération à la suivante.

La figure 3.7 donne un exemple d'évolution de lignes de niveau discrètes sur un tore selon la fonction de hauteur. Sur la figure 3.7(a), $\Gamma(v)$ est composé d'un seul contour discret. Il se divise ensuite en deux (figure 3.7(b)). Une bifurcation est donc créée dans le graphe (figure 3.7(e)). Les deux contours discrets fusionnent en un seul (figure 3.7(c)), ce qui crée une jonction dans le graphe (figure 3.7(f)).



(a) (b) (c) (d) (e) (f)

FIGURE 3.7 – *Evolution des lignes de niveaux discrètes et des contours discrets sur un tore selon la fonction de hauteur. Une bifurcation – (a) puis (b) – et une jonction – (b) puis (c) – sont détectées. Le graphe de Reeb se construit au fur et à mesure – (d), (e), (f)).*

La construction du graphe se faisant au fur et à mesure des itérations de l'algorithme de construction de lignes de niveau discrètes, et ces lignes ne se déconnectant pas dans les zones de points critiques de la fonction

d'application f_a (points rouges et noirs sur les modèles 3D de la figure 3.5), le graphe ne rend compte que des variations topologiques significatives.

Les détails concernant la construction des graphes de Reeb sont donnés dans la section 7 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document.

La figure 3.8 donne quelques exemples de graphes de Reeb calculés sur des maillages 3D de formes plus ou moins complexes.

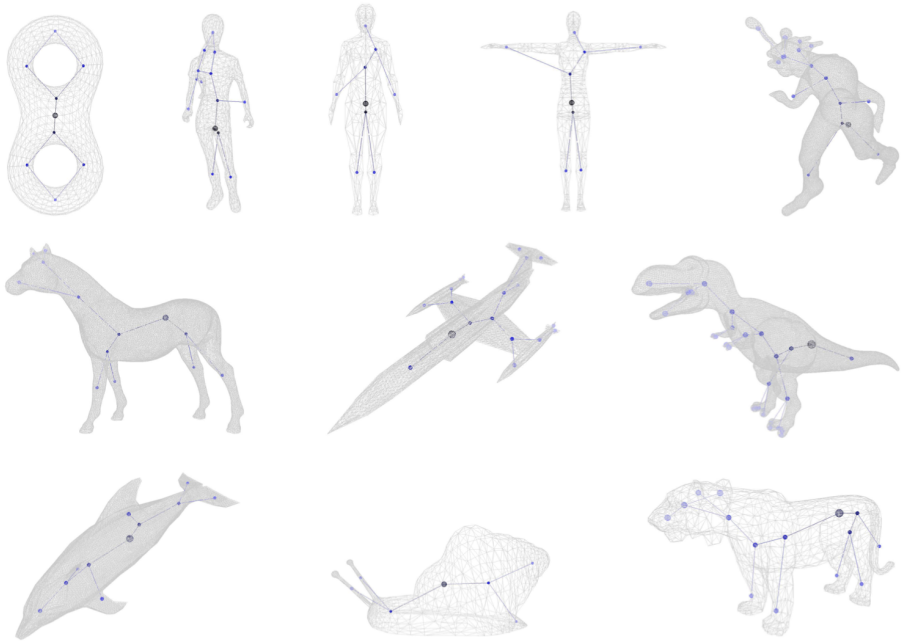


FIGURE 3.8 – *Exemples de graphes de Reeb de différents maillages 3D.*

3.2.8 Analyse géométrique des contours discrets

Les graphes de Reeb obtenus jusqu'à présent par notre méthode dénotent parfaitement la structure globale du modèle 3D qu'il représente. Cependant, un nœud du graphe représente une composante proéminente, quelque soit sa longueur. Par exemple, les graphes de Reeb des modèles humanoïdes de la figure 3.8 présentent des branches uniques pour chaque bras et chaque jambe.

Pour améliorer cela, nous nous sommes intéressés aux constrictions des contours discrets pour subdiviser plus finement certaines branches de longues composantes des modèles 3D. Une constriction peut être vue comme un resserrement puis un agrandissement d'un contour discret mettant ainsi en évidence une articulation ou une séparation entre deux par-

ties. L'idée générale est d'associer à chaque contour discret $\gamma(v)$ un indice de courbure et d'identifier les minima locaux de cet indice comme des constrictions au niveau desquelles un nœud de graphe est ajouté. L'indice utilisé ici est l'indice de courbure proposé par Koenderink et van Doorn (1992). Cet indice, calculé à partir des courbures principales, présente de bonnes propriétés d'invariance notamment aux transformations isométriques ainsi qu'à la mise à l'échelle uniforme. Nous avons aussi montré (Tierny et al. 2006) que cet indice de courbure donne des résultats plus probants que la courbure gaussienne ou la courbure moyenne.

Tous les détails concernant l'analyse géométrique des contours discrets sont donnés dans la section 8 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document.

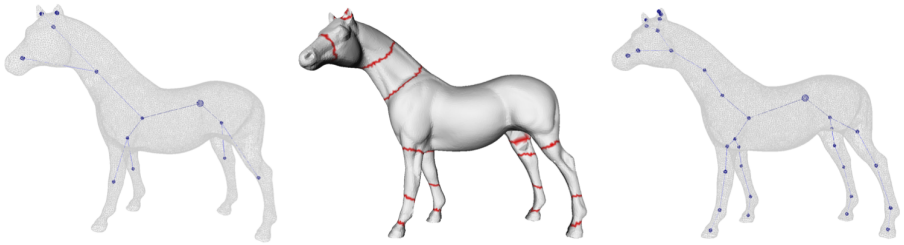


FIGURE 3.9 – *Graphe de Reeb classique, approximation des constrictions (en rouge), et graphe de Reeb amélioré d'un modèle 3D* cheval.

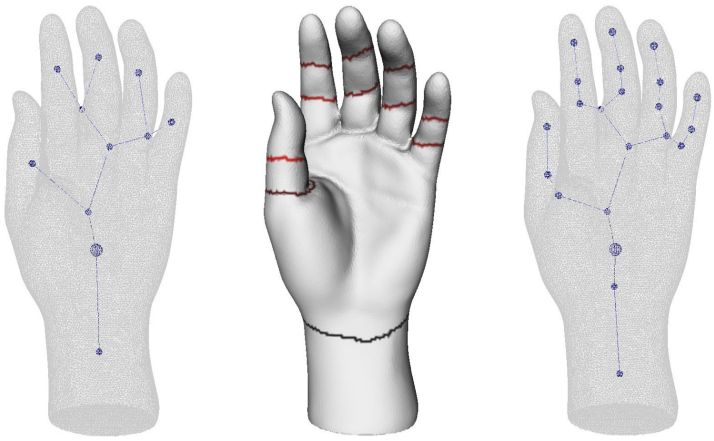


FIGURE 3.10 – *Graphe de Reeb classique, approximation des constrictions (en rouge), et graphe de Reeb amélioré d'un modèle 3D* main.

La figure 3.9 montre le graphe de Reeb classique d'un modèle 3D *cheval*, l'approximation des constrictions, et enfin le graphe de Reeb amélioré grâce à ces constrictions. On remarque que chaque jambe du cheval n'est représenté que par un seul et unique nœud dans le graphe de Reeb clas-

sique, alors qu'il est décomposé en quatre nœuds dans le graphe de Reeb amélioré.

La figure 3.10 montre le même type d'exemple mais avec un modèle 3D *main*. Là aussi, on voit clairement que les doigts sont plus détaillés et que chaque phalange est indiquée par un nœud dans le graphe amélioré.

3.2.9 Obtention d'un squelette topologique à partir d'un graphe de Reeb

Les graphes représentés à la figure 3.8 sont en fait des graphes duaux, c'est-à-dire que chaque composante connexe est représenté par un nœud – placé au barycentre de la composante – et leurs relations d'adjacence par une arête.

Pour obtenir un graphe de Reeb non dual, il suffit de placer un nœud, non pas au centre de chaque composante connexe, mais au centre de chaque contour discret. On a alors un squelette topologique comme le montrent les exemples de la figure 3.11.

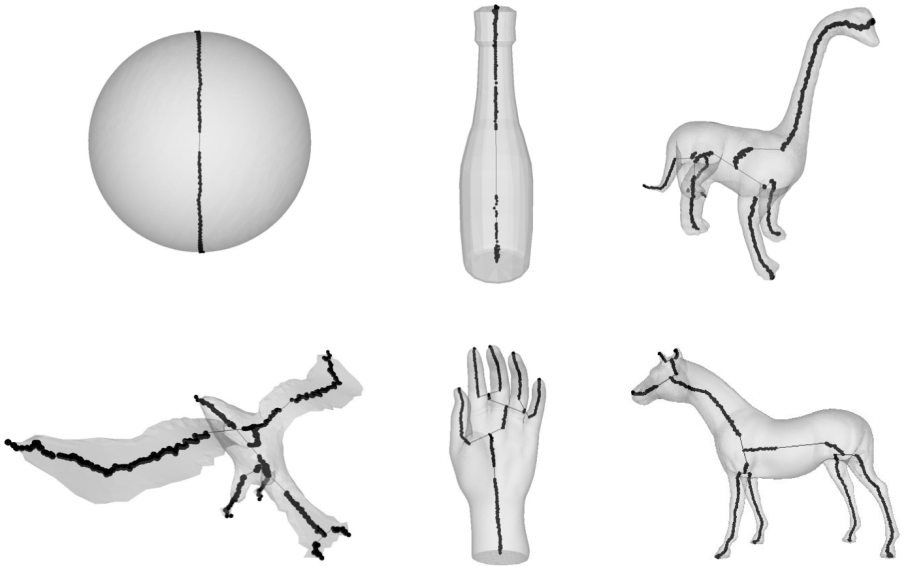


FIGURE 3.11 – *Exemples de squelettes topologiques obtenus à partir de graphes de Reeb améliorés.*

Tous les détails concernant le passage du graphe de Reeb au squelette topologique sont donnés dans la section 8 de la publication *Enhancing 3D mesh topological skeletons with discrete contour constrictions* (Tierny et al. 2008a) reproduite en annexe B page 156 du présent document. La figure 14 de cette publication donne des exemples de graphes de Reeb classiques puis de squelettes topologiques améliorés calculés grâce aux constrictions.

3.2.10 Complexité et temps de calcul

La complexité globale de notre algorithme est $O(n^2)$, avec n le nombre de points du maillage M représentant un objet 3D. Cette complexité correspondt en fait à l'étape de plus grande complexité de toutes les étapes, c'est-à-dire le calcul des contours discrets. La section 10.2 de la publication reproduite en annexe B page 156 du présent document donne de plus amples détails sur la complexité et les temps de calcul de notre méthode.

A titre indicatif, le tableau 3.1 donne quelques exemples de temps de calculs complets pour différents modèles sur un PC équipé d'un processeur de type Pentium 4 cadencé à 3 GHz et de 2 Go de mémoire vive. Ces durées sont nettement inférieures à celles d'autres méthodes de détection de constrictions (Hétroy 2005) et d'extraction de squelette (Wu et al. 2006) par exemple.

Modèle	Facettes	Sommets caract.	Constrictions	Durée approx.
Humanoïde	1 900	19	9	0.5 s.
Cheval 1	20 000	10	11	12 s.
Cheval 2	40 000	7	19	35 s.
Main 1	50 000	6	11	75 s.
Main 2	50 000	6	11	100 s.

TABLE 3.1 – *Durées d'exécution de notre algorithme sur différents modèles 3D.*

3.3 Trois applications

Le squelette topologique extrait d'un maillage 3D grâce à notre méthode nous donne de nombreuses indications sur la structure de l'objet représenté : composantes proéminentes (branches du graphe), trous (genre de la surface, grâce aux cycles dans le graphe), etc. Toutes ces informations sont alors utilisables dans de nombreux cadres applicatifs comme le soulignent Biasotti et al. (2003). Nous nous intéressons ici à trois d'entre eux : la manipulation de formes 3D à travers la déformation de maillages, l'indexation partielle de modèles 3D, et la segmentation de modèles 3D en parties porteuses de sens.

3.3.1 Déformation de maillages 3D

Puisque le squelette topologique amélioré est construit en explorant chaque point du maillage 3D, chaque nœud du squelette référence tous les points de la composante associée. De plus, comme le souligne Hétroy (2005), une constriction indique l'emplacement d'une articulation. En plaçant un point de contrôle du squelette à l'endroit des constrictions, et en permettant à un utilisateur de manipuler ces points et les composantes qui y sont attachés, il est possible d'"animer"[4] le maillage en le déformant au niveau des points de contrôle. D'autres stratégies plus évoluées sont également possibles (Lewis et al. 2000).

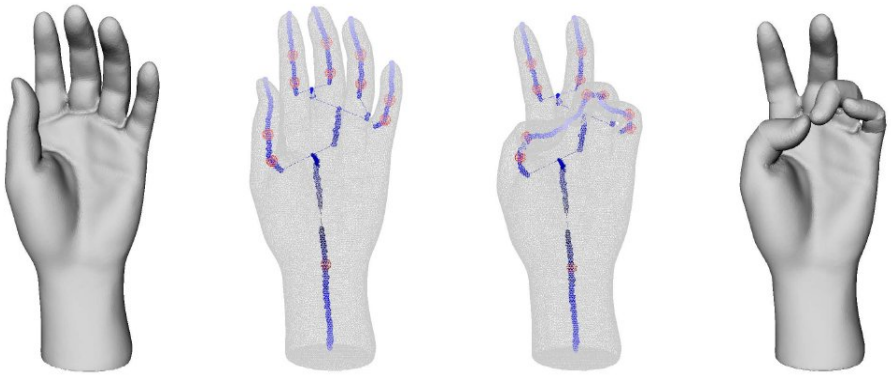


FIGURE 3.12 – *Exemple de déformation d'un maillage 3D. A gauche, le maillage original et son squelette avec ses points de contrôle. A droite, le squelette avec déformations aux articulations et le maillage déformé correspondant.*

La figure 3.12 montre un squelette topologique amélioré avec ses points de contrôle placés aux constrictions, ainsi que des déformations appliquées à celui-ci, et le maillage 3D résultant de ces déformations.

Une vidéo illustrant la déformation d'un maillage à partir de la manipulation d'un graphe amélioré est visible à l'URL suivante : `http://www.telecom-lille1.eu/people/vandeborre/hdr/chap2deformation.avi`

3.3.2 Indexation partielle de modèles 3D

Le deuxième domaine applicatif concerne l'indexation de maillages 3D, et notamment l'indexation partielle.

4. Ce terme est volontairement mis entre guillemets puisque des spécialistes de l'animation de maillages trouveraient, sans aucun doute, cette façon de faire par trop simpliste. Nous ne prenons par exemple pas en compte une éventuelle déformation non-rigide du maillage localement aux articulations.

Comme nous l'avons indiqué précédemment, chaque nœud du graphe – ou du squelette – référence tous les sommets de la composante qui lui est associée. Par conséquent, plutôt que de chercher un descripteur global pour le maillage 3D, il est possible de trouver des descripteurs locaux, un pour chaque composante mise en exergue par le graphe topologique. La combinaison de ces descripteurs donnera de meilleures performances qu'un simple et unique descripteur global. De plus, cette décomposition d'un maillage en parties locales définies chacune par un descripteur (Hilaga et al. 2001) est parfaitement adaptée à la description de maillages subissant des déformations non-rigides, comme l'articulation de personnages ou le pliage de formes.

Résumé de la méthode

Notre méthode s'appuie sur la paramétrisation de la surface des parties pour le calcul d'un descripteur de formes 3D insensible à la pose.

L'idée de base est la suivante : deux modèles 3D sont similaires si leurs sous-parties sont également similaires.

Nous avons appelé les descripteurs de sous-parties des *cartes de Reeb*, puisque ces sous-parties proviennent bien évidemment du graphe de Reeb amélioré présenté plus haut dans ce chapitre. Pour comparer les cartes de Reeb, nous procédons au *dépliage* de celles-ci dans un domaine plan canonique, puis nous comparons les distorsions produites par ces dépliages (signatures d'étirement) pour juger de la similarité des parties correspondantes. Enfin, pour juger la similarité globale entre deux modèles 3D, nous avons développé une méthode *brute force* qui consiste a calculer des distances entre toutes les parties des deux modèles et à apparier les parties dont les distances sont minimales tout en minimisant également la somme globale des distances (Tierny et al. 2007a), et une autre méthode, plus efficace, qui cherche des sous-motifs – que nous avons naturellement appelés *motifs de Reeb* – identiques entre les deux graphes de Reeb décrivant les objets (Tierny et al. (2009) – publication reproduite en annexe C page 175 de ce mémoire).

Nous donnons ensuite quelques résultats de similarité globale et partielle de forme en utilisant la méthode *brute force*. Les résultats utilisant la recherche de motifs de Reeb sont entièrement détaillés dans la section 5 de la publication *Partial 3D shape retrieval by Reeb pattern unfolding* (Tierny et al. 2009) reproduite en annexe C page 175 de ce mémoire.

Quelques résultats de similarité globale

Pour évaluer notre méthode, nous avons utilisé le jeu de données ISDB de l'Université de Tel-Aviv (Gal et al. 2007), composé de personnages et d'animaux articulés (106 modèles 3D au total répartis en 9 classes : chats, chevaux, chiens, dinosaures, grenouilles, humains, lions, loups et mains).

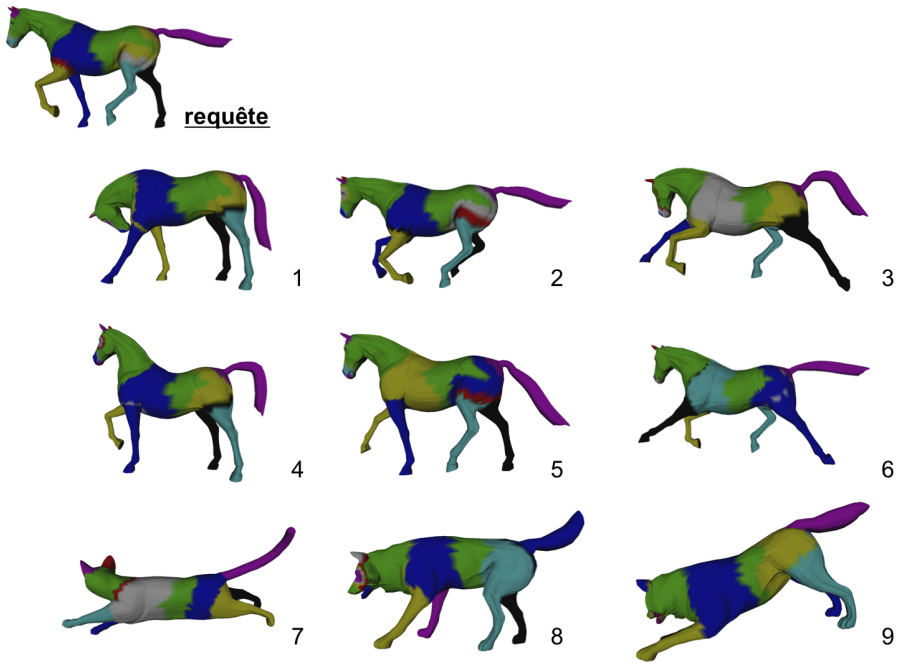


FIGURE 3.13 – *Requête et résultats dans l'ordre de pertinence. Les appariements de cartes de Reeb sont indiqués par des couleurs identiques.*

D'un point de vue qualitatif, la figure 3.13 montre une requête et les résultats retournés par ordre de similarité[5]. Les cartes de Reeb ont été mises en correspondance et coloriées de manière identique. A une exception près – la queue du loup en position 8 –, la queue du cheval requête a été mise en correspondance avec la queue de chacun des animaux retournés. Cet exemple montre de plus que la signature d'étirement est bel et bien robuste à la pose puisque des chevaux dans des poses différentes ont été retrouvés dans les premiers résultats.

D'un point de vue quantitatif, le tableau 3.2 donne des résultats et des comparaisons avec d'autres méthodes grâce à certains indicateurs de performance vus au chapitre précédent. La première ligne reporte les scores de notre algorithme de dépliage de cartes de Reeb. La deuxième ligne reporte les résultats du même algorithme mais en utilisant cette fois les

5. Le modèle-requête appartenant au jeu de données, le premier résultat est en fait le modèle-requête lui-même, mais cela cela ne présente pas d'intérêt ici. Ce premier résultat est donc ignoré

Méthodes	NN	FT	ST	E-Meas.	DCG
Tierny et al. (2007a)	**94.3 %**	**79.2 %**	**89.4 %**	**59.1 %**	**92.1 %**
Hilaga et al. (2001) attr.	88.7 %	70.6%	85.7 %	54.0 %	89.1 %
Funkhouser et al. (2003)	67.9 %	44.0 %	60.6 %	39.4 %	71.7 %

TABLE 3.2 – *Scores de performances des méthodes sur le jeu de données ISDB.*

attributs proposés par Hilaga et al. (2001) pour les sous-parties. La troisième et dernière ligne indique les scores de la méthode de Funkhouser et al. (2003). Les résultats de notre approche sont donc nettement supérieurs aux deux autres méthodes. Notons enfin que la méthode proposée par Gal et al. (2007) n'a pas pu être utilisée dans notre tableau car leurs résultats ont été obtenus sur un sous-ensemble non spécifié de 80 objets du jeu de données ISDB.

En terme de temps de calcul, une fois le graphe de Reeb du modèle 3D requête calculé, la comparaison entre la requête et l'ensemble du jeu de données ISDB s'effectue en environ 100 millisecondes sur un PC équipé d'un processeur de type Pentium 4 cadencé à 3 GHz et 2 Go de mémoire vive.

Notons que l'utilisation des motifs de Reeb donne des résultats encore meilleurs, notamment sur la base SHREC 2007 (Marini et al. 2007) pour laquelle notre méthode surpasse les méthodes présentées à ce concours. La section 5 de la publication *Partial 3D shape retrieval by Reeb pattern unfolding* (Tierny et al. 2009), reproduite en annexe C page 175 de ce mémoire, détaille ces résultats.

Quelques résultats de similarité partielle

Grâce à notre méthode, il est possible d'effectuer des appariements de sous-parties différentes. La figure 3.13 illustre bien cet avantage par le jeu de couleurs utilisé.

La figure 3.14 illustre également cette possibilité importante. La figure fait en effet un zoom sur les appariements des cartes de Reeb d'une main et d'un humanoïde. Les sous-parties correspondant aux doigts ont été mises en couleurs. Le pouce a été mis en correspondance correctement, et les autres doigts ont également été mis en correspondance malgré la différence de pose.

L'utilisation des motifs de Reeb donne, comme pour l'indexation globale, de meilleurs résultats de mise en correspondance comme le montrent

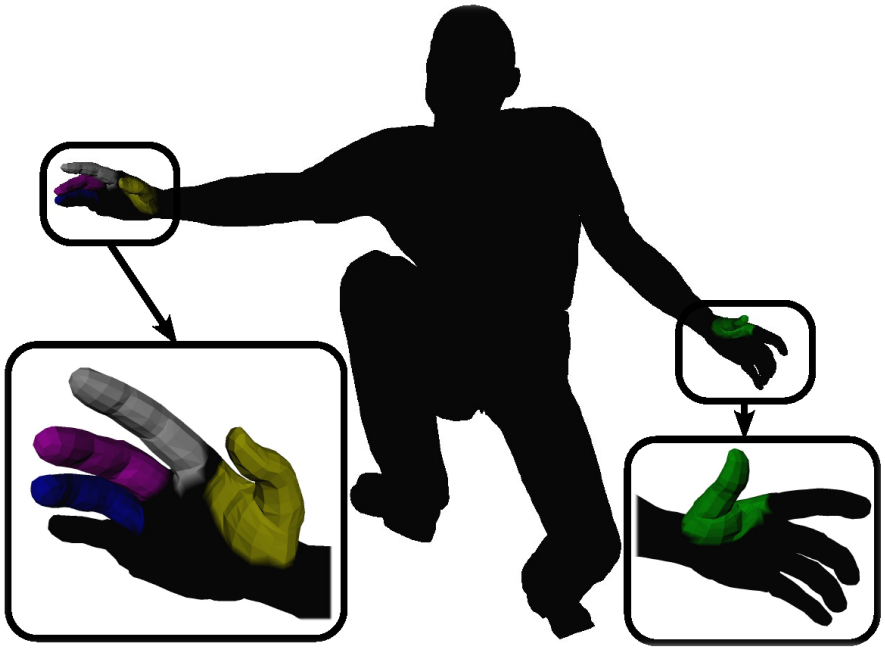


FIGURE 3.14 – *Appariements de cartes de Reeb entre une main et un humanoïde.*

les expériences de la publication reproduite en annexe C page 175 de ce mémoire

3.3.3 Segmentation de maillages 3D

Comme nous l'avons déjà souligné dans l'introduction général de ce mémoire, la segmentation est une tâche souvent essentiel à d'autres étapes de traitement de maillages 3D. Elle permet d'une certaine façon de mieux comprendre les formes, de donner un sens à une forme globale en la divisant en parties significatives.

La segmentation consiste à subdiviser un maillage surfacique en sous-parties – appelées également *régions*, *segments* ou *patches* – possédant des propriétés communes. Ces propriétés peuvent être strictement géométriques ou bien perceptuelles, voire sémantiques.

Une propriété importante de la segmentation de maillages, relevée notamment par Katz et al. (2005), est que deux surfaces représentant le même type d'objet soient segmentées de la même façon. Pour atteindre cet objectif, il est nécessaire de mener la segmentation de manière sémantique, et non simplement géométrique, en utilisant des notions de haut niveau comme celle définies dans la théorie de la perception humaines (Biederman 1987).

Nous avons donc proposé donc une méthode qui s'appuie sur les squelettes topologiques améliorés (Tierny et al. 2007b).

Résumé de la méthode

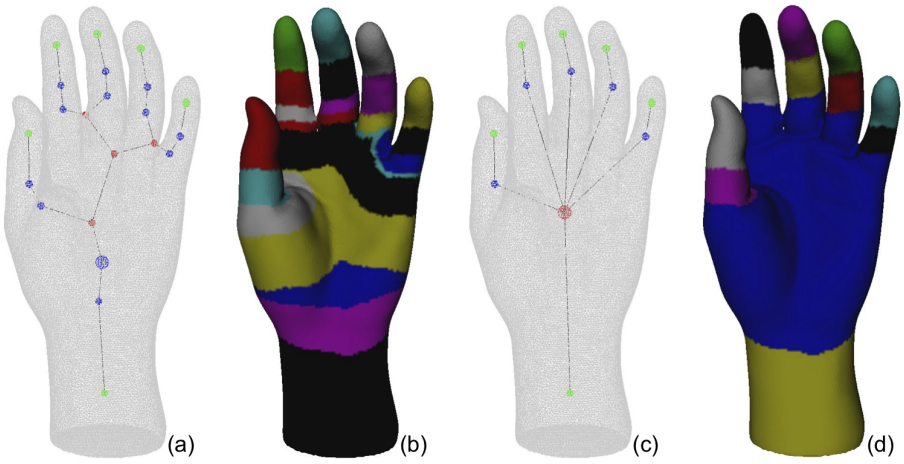


FIGURE 3.15 – *Etapes de la segmentation d'un maillage 3D.*

La première étape, illustrée par la figure 3.15(a), consiste à extraire le graphe amélioré du maillage (avec détection des constrictions) décrit dans ce chapitre. Les frontières entre les segments vont alors suivre les lignes de niveau définies par la fonction d'application. Afin d'être en accord avec l'idée intuitive que les frontières doivent suivre les courbures naturelles de la surface, la fonction d'application est légèrement modifiée pour intégrer, en plus de la distance géodésique, les courbures locales. Les lignes de niveau sont ainsi contraintes à suivre les parties concaves de la surface. En subdivisant le maillage selon les nœuds du graphe – qui correspondent chacun à un ensemble de sommets du maillage –, on obtient alors une segmentation brute (figure 3.15(b)). Cette segmentation brute est clairement une sur-segmentation au vu du nombre de segments non représentatifs détectés. Pour réduire le nombre de segments, un algorithme de fusion-jonction est mis en œuvre selon le voisinage de chaque segment. Le résultat de cette simplification est un graphe simplifié (figure 3.15(c)) qui donne finalement une segmentation fine et représentative dans laquelle, par exemple, la main est divisée en une paume et cinq doigts, eux-mêmes divisés en phalanges (figure 3.15(d)) .

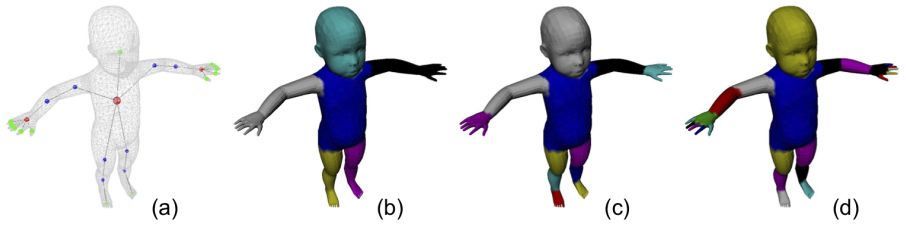


FIGURE 3.16 – *Segmentations hiérarchiques d'un maillage 3D selon les critères topologique.*

Segmentation hiérarchique

En suivant l'idée que la topologie d'un segment est plus importante que sa géométrie, nous avons développé une méthode de segmentation hiérarchique. Elle consiste à balayer le graphe simplifié depuis le nœud central (sphère rouge sur les figures 3.15(c) et 3.16(a)) et à subdiviser récursivement les branches du graphe selon les nœuds de cardinalité élevée.

La figure 3.16 illustre cette stratégie. A partir du graphe simplifié (figure 3.16(a)), la segmentation de premier niveau du modèle 3D *bébé* (figure 3.16(b)) est relativement grossière (tête, tronc et membres). Au niveau suivant (figure 3.16(c)), les membres sont subdivisés le long des branches (bras et main par exemple). Enfin, le dernier niveau est atteint (figure 3.16(d)) quand il n'est plus possible de subdiviser à nouveau les segments selon le graphe simplifié.

Quelques résultats

Le tableau 3.3 donne le temps de calcul pour obtenir la segmentation de plus haut niveau sur un PC équipé d'un processeur de type Pentium 4 cadencé à 3 GHz et de 2 Go de mémoire vive. Il indique également le nombre de facettes du maillage concerné et le nombre de segments au premier et au dernier niveau de hiérarchie de la segmentation. Notons également que la segmentation du *dinopet* s'effectue en 3,9 secondes avec notre méthode alors qu'elle prend 28 secondes avec la méthode de Katz et al. (2005).

La figure 3.17 montre les niveaux de segmentation de deux objets visuellement similaires : un cheval sans queue et un cheval avec sa queue, tous deux dans des résolutions de maillage différentes (40 000 facettes pour le premier, 5 000 pour le second). Notre algorithme suit donc bien l'idée intuitive que des maillages 3D représentant le même type d'objet doivent être segmentées de la même manière.

Modèles	Facettes	1er niveau	dernier niveau	Temps (s.)
Main	52 000	7	12	107
Bébé	10 000	6	24	5.4
Bitore	6 000	5	7	1.3
Cheval 1	40 000	6	21	34
Cheval 2	5 000	7	20	1.35
Dinopet	9 000	7	28	3.9

TABLE 3.3 – *Temps de calcul et nombre de segments au premier et au dernier niveau de hiérarchie de segmentation.*

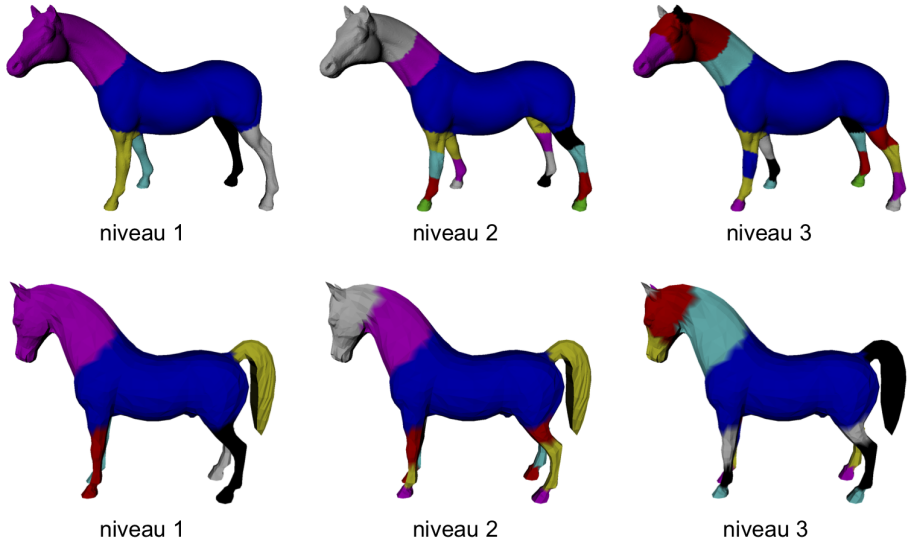


FIGURE 3.17 – *Segmentations cohérentes à chaque étape de deux objets similaires.*

3.4 Conclusion du chapitre

Dans ce chapitre, nous avons présenté nos travaux de recherche sur l'analyse de la structure de maillages 3D. Ces travaux correspondent à l'encadrement de la thèse de Julien Tierny. Leur objectif était de donner un certain sens aux modèles 3D maillés dont la structure de données est par nature – lot de points et de facettes en vrac – vide de sens. Ces travaux donnent des éléments de réponse à la deuxième question soulevée dans l'introduction générale de ce document : « Comment retrouver la structure d'un modèle 3D maillé sans connaissance *a priori* sur celui-ci ? »

Notre approche basée sur l'étude de la topologie des maillages 3D apporte, à notre avis, trois contributions principales. Nous avons tout d'abord mis au point un algorithme rapide et robuste d'extraction des sommets caractéristiques d'un maillage 3D. Nous avons ensuite montré que l'analyse topologique des contours discrets aboutit à un graphe de

Reeb significatif, c'est-à-dire qui n'encode pas de multiples détails inutiles. Enfin, nous avons ajouté une phase d'analyse géométrique des contours discrets pour détecter des constrictions sur les composantes proéminentes, permettant l'obtention d'un squelette amélioré, bien plus significatif visuellement et utilisable dans de nombreux domaines applicatifs.

D'un point de vue pratique, nous avons également mené diverses expériences et montré que notre approche était tout à fait valable pour la déformation automatique, l'indexation partielle, et la segmentation des maillages tridimensionnels.

Il convient cependant de nuancer l'efficacité et l'utilisabilité de notre approche. En effet, la fonction d'application – essentielle pour le calcul du graphe de Reeb – repose sur la distance géodésique entre les points du maillage. Si le calcul de cette distance est impossible, notre méthode devient bien évidemment totalement inapplicable. C'est par exemple le cas quand un modèle 3D est constitué de plusieurs composantes séparées (par exemple, la planète Saturne avec son anneau) ou quand le maillage n'est pas *manifold*. C'est notamment pour cette raison que nous n'avons pas pu utiliser la base du *Princeton Shape Benchmark*, référence dans le domaine de l'indexation 3D, pour nos expériences sur l'indexation partielle. De nombreux objets de cette collection ne sont en effet pas des 2-variétés ou sont composés de plusieurs parties non connexes.

De plus, si le modèle ne présente pas de protrusion évidente – comme le buste *Bimba Con Nastrino* de la figure 3.1 au début de ce chapitre – le graphe de Reeb sera trop simple, et même identique à celui d'une simple sphère, et par conséquent quasiment inutilisable pour une quelconque application.

Malgré ces limitations modérées, ces travaux nous ont ouvert la voie vers d'autres problèmes liés à la compréhension des maillages 3D et à l'apport de sens à ceux-ci. Un de ces problèmes est l'évaluation des méthodes de segmentation automatique des maillages 3D. En effet, lors de nos travaux sur ce sujet, nous nous sommes aperçus que les quelques contributions de ce domaine naissant ne cherchaient pas, ou très peu, à montrer leurs performances autrement que visuellement. Effectivement, il n'existait pas de critères objectifs pour évaluer et comparer les performances des méthodes entre elles. De plus, puisque nous nous intéressons au sens, n'y a-t-il pas moyen de prendre en compte des connaissances *a priori* sur les maillages 3D si elles existent ? Ces deux questions sont l'objet du chapitre suivant.

4 Segmentation de maillages 3D

La segmentation de maillages tridimensionnels est devenue un outil de pré-traitement nécessaire, pour ne pas dire indispensable, à de très nombreuses applications (reconnaissance de formes, compression, plaquage de textures, etc.) comme cela a été identifié par de nombreux auteurs (Sheffer et al. 2006, Zeckerberger et al. 2002).

La segmentation de maillages 3D consiste à subdiviser la surface représentée en différentes parties présentant des caractéristiques identiques. Les différentes parties présentent alors des caractéristiques communes, soit d'un point de vue strictement géométrique, soit d'un point de vue plus sémantique, et par conséquent porteur de sens. Les méthodes de segmentation automatique de maillages 3D sont également rangées dans ces deux catégories. Dans le cas des méthodes géométriques, les algorithmes s'appuient sur l'information géométrique de bas niveau, comme la courbure (Lavoué et al. 2005), pour définir des régions géométriquement homogènes. D'un point de vue sémantique, les algorithmes utilisent des notions de plus haut niveau comme on peut en trouver dans la théorie de la perception humaine (Biederman 1987). Ce genre d'approche, dont fait partie la méthode présentée à la fin du chapitre précédent (Tierny et al. 2007b), apporte souvent des décompositions plus significatives.

Le domaine, relativement récent, de la segmentation de maillages 3D est foisonnant – comme l'attestent les états de l'art de Attene et al. (2006b) et Shamir (2008) – et de nombreux auteurs continuent de proposer des méthodes. Cependant, jusqu'à il y a peu de temps, aucun outil automatique pour l'évaluation objective des méthodes n'a été proposé. Un tel outil doit apporter deux éléments indispensables : une collection de données

accompagnée de segmentations vérités-terrains, et une métrique de comparaison. De plus, la vérité-terrain apportant des connaissances *a priori* sur les maillages 3D, il est possible de s'appuyer sur ces connaissances pour proposer un nouvel algorithme de segmentation de maillages 3D.

Il est ici intéressant de noter deux choses. Premièrement, lorsque nous avons publié notre premier article sur l'évaluation automatique des segmentations et sur l'élaboration d'une collection de modèles 3D avec vérités-terrains servant à cette évaluation (Benhabiles et al. 2009), une autre équipe, de l'Université de Princeton aux États-Unis, a publié, quelques temps après, une approche dont la démarche et les idées sont totalement similaires aux nôtres (Chen et al. 2009). Deuxièmement, un an plus tard, alors que nous travaillions à la conception de notre algorithme de segmentation par apprentissage utilisant la vérité-terrain, une autre équipe, de l'Université de Toronto au Canada, a publié – cette fois peu de temps avant nous – une approche similaire à la nôtre (Kalogerakis et al. 2010). Ces deux méthodes, celle de Kalogerakis et al. (2010) et la nôtre (Benhabiles et al. 2011), sont, à ce jour et à notre connaissance, les deux seules méthodes de segmentation automatique de maillages 3D par apprentissage à partir d'une collection de segmentations vérités-terrains. Ceci renforce encore l'idée – s'il en est besoin – que l'évaluation des méthodes de segmentation de maillages 3D et la segmentation par apprentissage, sont deux domaines récents et innovants.

Dans ce chapitre, nous présentons les recherches que nous avons menées avec l'encadrement de la thèse d'Halim Benhabiles soutenue le 18 octobre 2011. Le chapitre se décompose de la façon suivante. Nous donnons en premier lieu une définition formelle de ce que nous entendons par *segmentation d'un maillage 3D*. Puis nous résumons les grandes étapes qui nous ont mené à l'élaboration du benchmark d'évaluation des segmentations. Nous nous attardons ensuite sur la constitution de la collection avec les segmentations vérités-terrains avant d'expliquer la métrique de comparaison que nous avons mise au point. Par la suite, nous utilisons cette nouvelle métrique pour comparer des algorithmes de segmentation automatique récents. Enfin, nous présentons une nouvelle méthode de segmentation de maillages 3D par apprentissage qui s'appuie sur la collection vérité-terrain.

4.1 Segmentation d'un maillage 3D

Comme nous l'avons déjà souligné dans l'introduction de ce chapitre, segmenter un maillage 3D consiste à le décomposer en différentes régions ayant des propriétés communes, soit d'un point de vue géométrique, soit d'un point de vue sémantique. Les régions ainsi définies sont également appelées *segments* ou *patches* en fonction de l'utilisation qui en sera faite.

Shamir (2008) définit la segmentation d'un maillage de manière plus formelle de la façon suivante.

Soient un maillage tridimensionnel M, tel que défini à la section 3.1.3 du chapitre 3 page 64 de ce document, et R l'ensemble des éléments du maillage – c'est-à-dire ou bien l'ensemble V des points (*vertices*), ou bien l'ensemble E des arêtes (*edges*), ou bien l'ensemble F des facettes (*faces*) du maillage M. Une segmentation S de M est un ensemble de sous-maillages $S = \{M_0, \ldots, M_{k-1}\}$ produit par le partitionnement de R en k sous-ensembles disjoints $R = \{R_0, \ldots, R_{k-1}\}$.

4.2 Pourquoi et par quels moyens évaluer des segmentations de maillages 3D ?

La segmentation de données multimédias n'est pas une idée nouvelle. Depuis de nombreuses années, des chercheurs se sont intéressés à la segmentation d'images 2D, puis de vidéos, et enfin de maillages 3D. Avec les méthodes de segmentation automatique est logiquement apparu le besoin d'évaluer les résultats obtenus par celles-ci.

Il est important d'évaluer les méthodes de segmentation pour plusieurs raisons : classer des algorithmes de segmentation entre eux, choisir l'algorithme qui donne les meilleurs résultats sur un certain type de données, analyser les résultats d'algorithmes pour éventuellement les améliorer, etc.

Il n'est cependant pas évident de fournir un tel outil d'évaluation qui soit à la fois objectif – *ie.* indépendant des données, de l'application et de ses utilisateurs – et quantitatif – *ie.* permettant de mesurer les performances. La qualité d'une segmentation (et donc la performance d'un algorithme de segmentation) dépend de l'utilisation qui en sera faite mais

également des utilisateurs pour qui la notion de "bonne" ou de "mauvaise" segmentation est très subjective.

Les premiers travaux d'évaluation de segmentation ont logiquement porté sur l'image 2D. Zhang et al. (2008) classent les méthodes d'évaluation de segmentation d'images en cinq groupes : les méthodes analytiques – qui s'intéressent à l'algorithme de segmentation lui-même (principe, complexité, etc.) –, les méthodes subjectives – qui s'appuient sur le jugement d'opérateurs humains –, les méthodes liées au système utilisateur – qui dépendent donc du système qui va utiliser les segmentations obtenues –, les méthodes empiriques avec qualité ou méthodes non-supervisées – qui jugent la qualité des segmentations obtenues selon des critères prédéfinies –, et les méthodes empiriques avec écart ou méthodes supervisées – qui mesurent l'écart entre des segmentations vérités-terrains et celles obtenues par les algorithmes de segmentation avec une métrique de comparaison.

Les méthodes analytiques ont l'inconvénient de ne s'intéresser qu'à l'algorithme et non au résultat final. Les méthodes subjectives, comme leur nom l'indique, dépendent d'observateurs humains pour chaque étape de l'évaluation et ne peuvent pas être intégrées dans un système automatique. Les méthodes liées au système utilisateur des segmentations dépendent fortement de ce dernier et ne sont donc pas généralisables. Les méthodes non-supervisées dépendent de critères définis *a priori* et ne permettent pas d'évaluer quantitativement les segmentations. Enfin, les méthodes supervisées, bien qu'elles dépendent également d'opérateurs humains, sont les plus représentées (Martin et al. 2001, Unnikrishnan et al. 2007) car elles sont automatisables et car elles assurent une évaluation quantitative grâce à la métrique de comparaison (calcul de l'écart entre segmentations vérités-terrains et segmentations obtenues) ainsi qu'une évaluation – relativement – subjective grâce à la vérité-terrain.

Pour l'évaluation de segmentation de maillages 3D, outre la très récente méthode de Chen et al. (2009), publiée peu de temps après la nôtre (Benhabiles et al. 2009) et sur laquelle nous reviendrons dans les sections suivantes, quelques tentatives ont été faites par Attene et al. (2006b) et Berretti et al. (2009). Dans leur état de l'art, Attene et al. (2006b) proposent des critères d'évaluation tels que l'aspect des frontières entre segments, les propriétés hiérarchiques et multi-échelle, la complexité, etc. Mais ces critères ne permettent pas une évaluation objective et mesurable de la qualité des segmentations. Berretti et al. (2009) s'appuient quant à eux sur une col-

lection de segmentations vérités-terrains. Mais celle-ci n'est pas accessible publiquement et ne comporte que quelques modèles 3D très simples.

Enfin, une réflexion générale a été menée par Delest et al. (2008) sur l'évaluation des méthodes de segmentation de maillages 3D et a montré l'intérêt de la constitution d'une collection de segmentations vérités-terrains dans ce but.

La méthode d'évaluation des segmentations 3D que nous proposons dans ce chapitre appartient à la catégorie des méthodes supervisées. Dans les deux prochaines sections de ce chapitre, nous présentons notre collection avec segmentations vérités-terrains puis la métrique de comparaison, utilisées pour l'évaluation quantitative et objective des méthodes de segmentation de maillages 3D.

4.3 Collection de modèles 3D avec segmentations vérités-terrains

Une méthode d'évaluation supervisée (ou méthode empirique avec écart) nécessite en premier lieu la connaissance de ce que l'on pourrait appeler une *bonne* segmentation pour chaque maillage tridimensionnel, ou au moins d'une ou plusieurs segmentations que la plupart des gens juge *correctes*.

Mais qu'est-ce qu'une segmentation *correcte* ? La réponse a cette question est bien entendu très subjective. Elle dépend de ce que l'on souhaite faire de la segmentation finale et donc de l'application utilisée par la suite (par exemple le placage de textures (Sander et al. 2001) ou l'imagerie médicale (Gerig et al. 2001)). Elle dépend également du type de modèles 3D à segmenter (par exemple les objets CAO (Lavoué et al. 2005)). Enfin, elle dépend d'une éventuelle spécialisation des gens en charge de créer ces segmentations.

Nous avons choisi de répondre à cette question en nous appuyant sur le "bon sens commun" pour juger des parties significatives des modèles 3D. Pour chaque maillage 3D, nous considérons plusieurs segmentations manuelles, fruit du travail d'une quarantaine de volontaires non spécialistes. Nous appelons cette collection de maillages 3D accompagnés de leurs segmentations manuelles *la collection vérité-terrain* [1].

1. La collection vérité-terrain ainsi que l'ensemble du benchmarck (outils) sont accessibles en ligne. Voir la section 4.5.3, page 106 de ce chapitre pour de plus amples détails.

4.3.1 Constitution de la collection vérité-terrain

La collection vérité-terrain est constituée de vingt-huit modèles 3D maillés classés en cinq catégories : animaux, meubles, mains, humanoïdes et bustes. Ces modèles, colorés selon une de leurs segmentations vérités-terrains, sont représentés à la figure 4.1.

FIGURE 4.1 – *Les vingt-huit modèles 3D de la collection vérité-terrain. Chaque modèle est coloré en fonction d'une de ses vérités-terrains.*

Tous ces objets sont relativement communs et variés afin, d'une part, de ne pas nécessiter de connaissances spécialisées de la part des volontaires, et d'autre part, de ne pas privilégier certains algorithmes spécia-

lisés dans la segmentation d'un type d'objet particulier au moment de l'évaluation.

Comme nous l'avons souligné précédemment, la correction d'une segmentation est une question subjective. Pour résoudre – ou tout du moins atténuer – ce problème, nous associons à chacun des vingt-huit modèles 3D de la collection quatre segmentations vérités-terrains, pour un total de cent-douze segmentations vérités-terrains. Chacune d'entre elles est le travail d'un volontaire non spécialiste. Ainsi des différences de point de vue sur la manière de segmenter un modèle 3D sont prises en compte dans notre collection. La figure 4.2 montre deux segmentations manuelles d'un modèle *cheval* obtenues par deux volontaires. Les niveaux de détails sont totalement différents : là où le premier volontaire a segmenté de manière très précise, par exemple, les membres du cheval en trois parties distinctes, le second volontaire a préféré segmenter en parties plus générales. De manière générale, tous les volontaires ont segmenté les modèles 3D en suivant le "bon sens commun". Aucune segmentation "bizarre" ou défiant ce fameux bon sens n'a été observée.

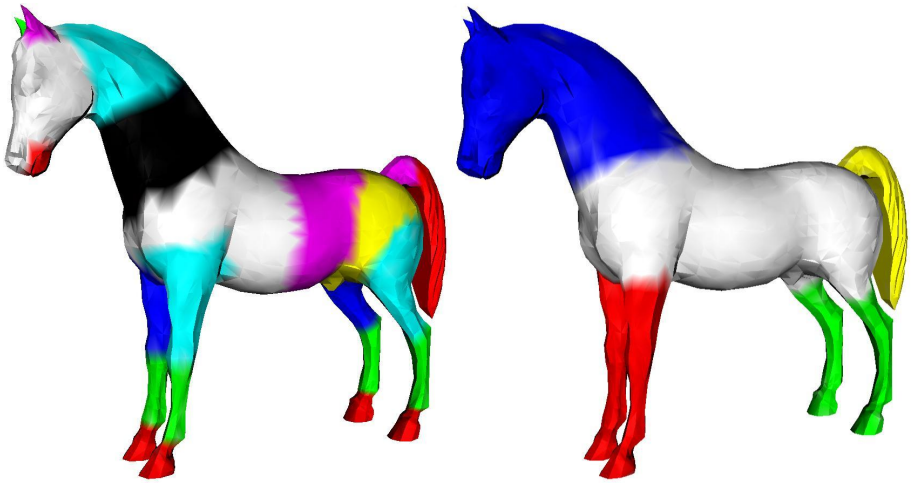


FIGURE 4.2 – *Deux segmentations vérités-terrains pour un modèle de cheval vues par deux volontaires différents.*

Enfin, pour être certain que les modèles seront pris en compte par la plupart des algorithmes de segmentation, tous les modèles 3D de la collection vérité-terrain sont des maillages *manifolds* (2-variétés). Ils sont extraits de bases de données publiques comme la base GAMMA de l'INRIA [2], le Princeton Shape Benchmark [3] (Shilane et al. 2004) et le AIM@SHAPE Shape Repository [4].

2. http://www-roc.inria.fr/gamma/gamma.php
3. http://shape.cs.princeton.edu/benchmark/
4. http://shapes.aimatshape.net/

4.3.2 Protocole de segmentation manuelle

Les segmentations ont été réalisées grâce au logiciel libre *MeshLab*[5]. MeshLab est un logiciel de visualisation, de traitement, d'édition, etc. de modèles 3D. Nous avons plus particulièrement utilisé l'outil *virtual brush* afin de colorier les frontières entre segments, puis utilisé un algorithme de propagation de couleurs pour colorier totalement le modèle 3D et obtenir ainsi une segmentation finale où chaque partie significative (segment) est représenté par une couleur.

Chaque segmentation vérité-terrain est le travail d'un volontaire. Les volontaires venant de divers horizons (étudiants, enseignants, administratifs, etc.), le niveau d'utilisation d'un logiciel tel que MeshLab peut être un frein à ce travail. Nous avons par conséquent assisté les volontaires dans cette tâche. Nous avons donc – plus ou moins – supprimé l'éventuelle appréhension liée à l'utilisation d'un tel logiciel, obtenu des segmentations précises sur les indications des volontaires, et accéléré ce travail fastidieux. Nous n'avons cependant imposé aucune condition sur la manière de segmenter les modèles, que ce soit en nombre de parties ou sur les endroits où effectuer la séparation entre celles-ci. Les volontaires sont restés libres dans la manière de mener à bien la segmentation des modèles 3D qui leur étaient confiés.

Plus de détails sur la constitution de la collection vérité-terrain sont donnés dans notre publication à *IEEE International Conference on Shape Modeling and Applications (SMI)* en 2009 (Benhabiles et al. 2009).

4.3.3 Comparaison avec la collection de l'Université de Princeton

Comme nous l'avons indiqué dans l'introduction de ce chapitre, l'Université de Princeton aux Etats-Unis (Chen et al. 2009) propose une collection complémentaire à la nôtre. Elle est disponible en ligne[6].

Quelques différences sont à noter. Tout d'abord, la collection de l'université de Princeton comporte 380 objets extraits de la base *Watertight Models Track*[7] du concours SHREC 2007 (Giorgi et al. 2007), là où notre collection n'en compte que vingt-huit, collectés de plusieurs sources publiques, pour permettre une évaluation rapide. Pour réaliser les segmentations vérités-terrains, l'Université de Princeton a choisi l'application web

5. http://meshlab.sourceforge.net/
6. http://segeval.cs.princeton.edu/
7. http://watertight.ge.imati.cnr.it/

Amazon Mechanical Turk[8] sans aucune supervision auprès des volontaires. Nous avons fait le choix d'aider les volontaires pour obtenir des segmentations plus précises. Enfin, les segmentations de Princeton sont basées facettes, alors que les nôtres s'appuient sur les sommets du maillage.

Les 380 objets 3D de la collection de l'université Princeton sont répartis en 19 catégories contenant chacune 20 objets. Chaque objet est accompagné de plusieurs vérités-terrains. La figure 4.3 montre un exemple d'objet pour chaque catégorie de la collection. Les lignes sombres représentent la moyenne des frontières entre segments. Plus le trait est sombre, plus les volontaires ont choisi cette ligne comme frontière.

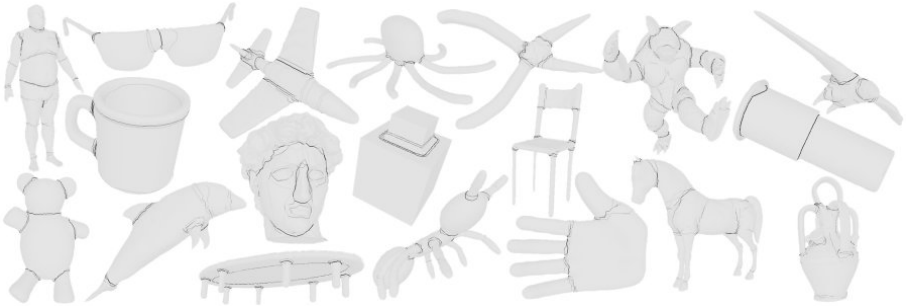


FIGURE 4.3 – *Exemples d'objets 3D extraits de la collection de l'université de Princeton avec leurs vérités-terrains sous forme de frontières (illustration extraite de (Chen et al. 2009)).*

Nous utiliserons cette collection, en complément de la nôtre, dans la suite de ce chapitre pour l'évaluation des méthodes de segmentation automatique de maillages 3D. Nous l'utiliserons également comme base d'apprentissage pour notre nouvel algorithme de segmentation présenté à la fin de ce chapitre.

4.4 Métriques de comparaison de segmentations

Après la constitution d'une collection vérité-terrain, il est nécessaire de posséder une métrique de comparaison des segmentations de maillages 3D entre elles. Cette métrique servira à comparer les segmentations obtenues par les algorithmes de segmentation automatique aux segmentations verités-terrains de la collection. En appliquant cette métrique à l'ensemble des modèles de la collection vérité-terrain, on obtiendra alors une évaluation des performances de l'algorithme de segmentation en question, soit de manière globale, soit classe d'objet par classe d'objet, rendant alors

8. `http://www.mturk.com`

possible la comparaison avec d'autres algorithmes de segmentation automatique.

Nous nous intéressons tout d'abord aux propriétés que doit posséder une métrique de comparaison fiable, puis nous présentons quelques métriques de la littérature réparties en trois catégories, avant d'introduire notre nouvelle métrique et sa version normalisée.

4.4.1 Propriétés d'une métrique de comparaison fiable

Pour être fiable, une métrique de comparaison de segmentations doit posséder les propriétés suivantes :

- **Gestion des cas dégénérés.** Le résultat de la métrique doit être proportionnel au degré de similarité entre une segmentation automatique et une segmentation vérité-terrain d'un même modèle 3D. Par exemple, une sur-segmentation, *ie.* une segmentation *dégénérée* dans laquelle chaque point (ou facette) représente un segment – comparée à n'importe quelle segmentation vérité-terrain doit avoir un score de métrique très mauvais.
- **Tolérance au raffinement.** Une métrique de comparaison doit être tolérante au niveau de raffinement des segmentations d'un même objet. En effet, comme l'illustre la figure 4.2 page 93, pour un même objet, on peut avoir une segmentation fine (ou détaillée) et une autre plus grossière. Néanmoins, les deux segmentations sont cohérentes.
- **Indépendance à la cardinalité.** La métrique doit être capable de comparer deux segmentations qui n'ont pas le même nombre de segments ou des tailles de segments différentes.
- **Tolérance à l'imprécision des frontières de segments.** Une métrique doit être tolérante au placement des frontières entre segments car celles-ci peuvent varier de quelques points (ou facettes) sur le maillage entre deux segmentations sans pour autant changer la cohérence de celles-ci.
- **Prise en compte des vérités-terrains multiples.** Les vérités-terrains multiples pour un même objet 3D permettent, comme nous l'avons déjà indiqué, de prendre en compte la subjectivité des segmentations vérités-terrains réalisées par les volontaires. La métrique doit prendre cela en compte.
- **Comparaisons significatives.** La métrique doit être en mesure d'indiquer la meilleure segmentation parmi de nombreuses autres seg-

mentations d'un même objet (qualité de la segmentation), mais aussi quel type d'objet est le plus commode à segmenter par un algorithme de segmentation automatique.

4.4.2 Quelques métriques de comparaison de segmentations

Nous résumons ici quelques métriques de comparaison de segmentations de maillages 3D avant de présenter notre nouvelle métrique.

Les métriques existantes sont classées en trois catégories, similaires à celles proposées par Unnikrishnan et al. (2007) pour l'image 2D : métriques par correspondance de frontières, métriques par différence de régions, et métriques basées sur des tests non-paramétriques.

Tous les détails concernant ces trois catégories et les métriques associées sont donnés dans la section 2.2 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 de ce mémoire. Nous ne présentons ici qu'un rapide résumé de ces catégories.

Métriques par correspondance de frontières. Ce type de métrique s'intéresse aux frontières entre les régions de deux segmentations.

Chen et al. (2009) proposent une telle métrique nommée *Cut Discrepancy* (CD). Cette métrique mesure la distance entre les frontières de segments les plus proches de deux segmentations. Une valeur de $CD(S_1, S_2) = 0$ indique une égalité stricte entre les deux segmentations S_1 et S_2. Comme le soulignent les auteurs, la métrique n'est pas définie quand une segmentation ne possède pas de frontière (quand il n'y a qu'un seul et unique segment). Elle ne gère donc pas les cas dégénérés. La métrique tend vers 0 quand la différence du nombre de segments entre deux segmentations augmente. Elle n'est donc pas tolérante au raffinement. La métrique ne permet de prendre en compte qu'une seule vérité-terrain.

Métriques par différence de régions. Ce type de métrique s'intéresse à la similitude entre les régions de deux segmentations.

Berretti et al. (2009) proposent un index nommé *Overlap Index* (OI) qui représente le recouvrement entre chaque segment de deux segmentations. L'indice global $O(S_1, S_2)$ étant la moyenne des index sur tous les segments de S_1. Une valeur de 1 signifie que les deux segmentations sont strictement

identiques, sinon la valeur est inférieure à 1. Cette métrique ne gère pas les cas dégénérés. Elle ne permet pas non plus la comparaison à des vérités-terrains multiples.

Dans la même catégorie, Chen et al. (2009) et nous-mêmes (Benhabiles et al. 2009) avons généralisé la mesure d'erreur de cohérence, utilisée pour l'évaluation de la segmentation d'images 2D par Martin et al. (2001), à l'évaluation de la segmentation de maillages 3D. Cette mesure utilise une erreur de raffinement local à chaque point (ou facette, selon l'élément utilisé pour segmenter les maillages) des segmentations comparées. Les erreurs locales sont alors combinées de deux façons : *Global Consistency Error* (GCE) et *Local Consistency Error* (LCE). Une valeur de 0 pour l'une ou l'autre des métriques indique que les deux segmentations comparées sont identiques. Une valeur de 1 indique la divergence la plus grande. Ces mesures prennent en compte le raffinement, dans un seul sens pour le GCE, dans les deux sens pour le LCE. Ces mesures ne gèrent pas les cas dégénérés (quand, par exemple, S_1 compte un segment par point, et S_2 ne comporte qu'un seul segment, les deux mesures ont une valeur de 0).Elles ne prennent pas non plus en compte les vérités-terrains multiples.

Enfin, et toujours dans la même catégorie, Chen et al. (2009) utilisent une autre mesure nommée *Hamming Distance* (HD). La distance de Hamming entre deux segmentations mesure la différence de régions entre leurs ensembles respectifs de segments. Une valeur de métrique de 0 indique une égalité parfaite entre les deux segmentations comparées. Comme l'indiquent les auteurs, cette mesure fonctionne bien si la correspondance entre les segments se fait facilement. Dans le cas contraire, elle n'est pas significative. Enfin, comme les précédente métriques, elle ne permet pas la comparaison à des vérités-terrains multiples.

Métriques basées sur des tests non-paramétriques. Ces métriques sont adaptées de tests non-paramétriques utilisées pour l'évaluation des méthodes de classification comme le *Rand Index* (Rand 1971).

Chen et al. (2009) proposent une adaptation du *Rand Index* à l'évaluation des segmentations de maillages 3D. Cette méthode considère la comparaison de deux segmentations avec des nombres de segments différents comme un problème de mise en relation d'ensembles par leurs étiquettes (*labels*). Elle considère en fait le ratio d'éléments ayant la même étiquette dans les deux segmentations. Cette métrique ne permet pas la comparaison à des vérités-terrains multiples.

Pour conclure, on note qu'aucune des métriques résumées ici ne respecte toutes les propriétés définies dans la section 4.4.1 de ce chapitre.

4.4.3 Une nouvelle métrique de comparaison de segmentations

Afin de respecter les propriétés énoncées à la section 4.4.1 de ce chapitre, et notamment la prise en compte des verités-terrains multiples, nous proposons une nouvelle métrique, et sa version normalisée, de comparaison de segmentations de maillages 3D.

Nous nous sommes appuyés sur les travaux de Unnikrishnan et al. (2007) concernant l'évaluation de la segmentation des images 2D. Ces auteurs proposent une interprétation probabiliste du *Rand Index* (Rand 1971) et en ont montré les performances.

3D Probabilistic Rand Index – 3D-PRI

Pour la comparaison de deux segmentations d'images 2D, le *Rand Index* original considère chaque paire de pixels entre les deux segmentations ayant la même étiquette (*ie.* qui appartiennent au même segment) et calcule le ratio du nombre de ces paires par rapport au nombre total de paires possibles. Un score de 1 indique que les deux segmentations sont parfaitement identiques, un score de 0, une dissimilarité totale (quand une segmentation comporte un segment par pixel, et l'autre un seul et unique segment).

L'interprétation probabiliste du *Rand Index* d'Unnikrishnan et al. (2007), nommée *Probabilistic Rand Index* (PR Index), est une amélioration du *Rand Index* pour comparer un segmentation à tester avec un ensemble de segmentations vérités-terrains. Ce nouvel index considère non plus de simples appariements d'étiquettes entre pixels mais une pondération des paires de pixels entre la segmentation à tester et un ensemble de segmentations vérités-terrains, en fonction de la variabilité de ces dernières. Le score du PR Index s'interprète de la même façon que celui du *Rand Index* : 1 quand la segmentation à tester est identique à toutes les segmentations vérités-terrains, *O* en cas de dissimilarité totale.

L'adaptation du PR Index de la segmentation d'images 2D à la segmentation de maillages 3D est très simple puisqu'elle consiste à utiliser les sommets (*vertices*) des maillages 3D à la place des pixels des images 2D.

Puisque la segmentation à tester et l'ensemble des segmentations vérités-terrains concernent le même maillage 3D, la correspondance sommet à sommet entre les segmentations est immédiate. Nous avons nommé cette métrique tout simplement *3D Probabilistic Rand Index* (3D-PRI).

Tous les détails concernant le 3D-PRI (labels, variabilité des labels dans les segmentations-vérités terrains, etc.) sont donnés dans la section 3 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 de ce mémoire.

3D Normalized Probabilistic Rand Index – 3D-NPRI

Bien que la métrique 3D-PRI possède les bonnes propriétés citées à la section 4.4.1 de ce chapitre (démontré par Unnikrishnan et al. (2007) pour le PR Index et donc parfaitement valable pour le 3D-PRI), elle souffre d'un handicap important : la faible variation de ses valeurs et par conséquent son faible pouvoir discriminant. De plus, l'interprétation des valeurs basses et élevées n'est pas toujours évidente.

Pour remédier à cela, nous proposons, de la même manière qu'Unnikrishnan et al. (2007), une version normalisée du 3D-PRI nommée *3D Normalized Probabilistic Rand Index* (3D-NPRI).

Une méthode classique de normalisation d'index (Fowlkes et Mallows 1983) par rapport à une référence pour une comparaison de scores plus efficace est la suivante :

$$Normalized\ index = \frac{Index - Expected\ index}{Maximum\ index - Expected\ index}$$

Comme le soulignent Unnikrishnan et al. (2007), la valeur du *Maximum index* est généralement fixée à 1, c'est-à-dire la valeur maximale du PR Index ou du 3D-PRI dans notre cas.

L'estimation de l'*Expected index* est plus complexe. Unnikrishnan et al. (2007) estiment cette valeur à l'aide de l'ensemble des segmentations vérités-terrains de toutes les images de la collection, en les comparant entre elles, pixel à pixel. Cette façon de faire est tout à fait possible, bien que très longue mais améliorable par un échantillonnage uniforme des pixels, pour des images 2D de résolutions identiques. Par contre, elle devient impossible dans le cas de maillages 3D où les modèles de la collection possède des définitions différentes, en terme de nombre de sommets

et de facettes ou en terme de connectivité. Pour évaluer correctement cet *Expected index* sans imposer de contraintes intenables sur la collection de maillages 3D – même nombre de sommets et de facettes et même connectivité pour tous les maillages –, nous introduisons des segmentations aléatoires pour chaque maillage de la collection.

Tous les détails concernant le 3D-NPRI (estimation de l'*Expected index*, segmentations aléatoires, etc.) sont donnés dans la section 4 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 du présent document.

4.4.4 Propriétés de la métrique 3D-PRI / 3D-NPRI

Pour mettre en évidence les bonnes propriétés que possède notre métrique, nous avons mené des expériences de comparaison avec les métriques introduites à la section 4.4.2.

Tous les détails de ces expériences sont donnés dans la section 6 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 du présent document.

Comparaison avec les métriques existantes

Nous résumons ici les résultats obtenus des comparaisons des métriques pour vérifier les propriétés énoncées dans la section 4.4.1 de ce chapitre.

- **Gestion des cas dégénérés.** Nous avons utilisé trois niveaux de segmentation aléatoires en fonction du nombre de segments (de 2 ou 3 segments à plus d'une cinquantaine) et avons observé le comportement des métriques. Puisque les segmentations sont générées aléatoirement, les scores de métriques doivent être très bas. Seule notre métrique répond à ce critère et gère donc les cas dégénérés. Détails de ces expériences à la section 6.1 de l'article reproduit en annexe D de ce document.
- **Tolérance au raffinement.** Nous avons mené deux expériences pour cette propriété. La première utilise deux raffinements mutuels parfaits d'un modèle 3D. Toutes les métriques, à l'exception de *Cut Discrepancy*, présentent un bon comportement et proposent un score

proche de 1. La deuxième expérience utilise des segmentations hiérarchiques issues de l'algorithme d'Attene et al. (2006a). Les résultats montrent les excellentes performances des métriques *Global* et *Local Consistency Error*. Notre métrique donne des résultats très légèrement inférieurs à ces derniers. Détails de ces expériences à la section 6.2 de l'article reproduit en annexe D de ce mémoire.

- **Indépendance à la cardinalité.** Des deux précédentes séries d'expériences, on déduit que seule la métrique *Cut Discrepancy* est sujette aux problèmes de cardinalité. Les autres métriques se comportent bien. Voir la section 6.3 de l'article reproduit en annexe D pour plus de détails.
- **Tolérance à l'imprécision des frontières de segments.** Pour cette propriété, nous avons simplement édité manuellement une segmentation d'un objet simple (un bitore) pour obtenir plusieurs segmentations où seule la frontière entre les deux segments de l'objet diffère très légèrement. Toutes les métriques, à l'exception, une fois de plus, *Cut Discrepancy*, se comportent correctement face à ce problème. Détails à la section 6.4 de l'article reproduit en annexe D de ce document.
- **Prise en compte des vérités-terrains multiples.** Par sa conception probabiliste, la métrique 3D-PRI / 3D-NPRI prend en compte les vérités-terrains multiples.
- **Comparaisons significatives.** Nous avons cette fois utilisé des segmentations produites par l'algorithme de Tierny et al. (2007b) et les avons comparées à notre collection vérité-terrain. Seule la métrique 3D-NPRI (donc version normalisée) rend parfaitement compte de la qualité des segmentations de Tierny et al. (2007b) par rapport à la collection vérité-terrain. Détails à la section 6.5 de l'article reproduit en annexe D de ce document.

Expérience subjective et pouvoir discriminant de notre métrique

Pour prouver le pouvoir discriminant de notre métrique normalisée 3D-NPRI, et également l'efficacité de notre collection vérité-terrain, nous avons mené une expérience subjective. Dans cette expérience, des observateurs ont évalué subjectivement la qualité d'un ensemble de segmentations. Ces segmentations proviennent de trois sources : notre collection vérité-terrain, des algorithmes de segmentation automatique (Attene et al. (2006a), Lavoué et al. (2005), Shapira et al. (2008) et Tierny et al. (2007b))

et des segmentations aléatoires. Au total, 250 segmentations ont été créées à partir de notre collection de 28 modèles 3D.

Protocole subjectif. Pour le protocole, nous nous sommes inspirés de ce qui se fait en évaluation de qualité d'image (Rogowitz et Rushmeier 2001), de vidéo (Gelasca et al. 2006), etc. Après une phase d'explication orale et une rapide phase d'entrainement – pendant laquelle on montre une segmentation vérité-terrain et une segmentation aléatoire afin de définir ce qui doit être considéré comme une excellente segmentation et comme une très mauvaise –, vient la phase finale de notation. A cette étape, l'observateur note, selon son propre jugement, 50 segmentations par une note de 1 (très mauvaise segmentation) à 10 (excellente segmentation). Cinquante observateurs ont participé à l'expérience, pour un total de dix notes par segmentation. Une note d'opinion moyenne (*Mean Opinion Score*) est alors calculée pour chaque segmentation.

Cohérence des notations. Le coefficient de corrélation intra-classe (*Intra-class Correlation Coefficient* – ICC), qui évalue la variation des scores donnés par les observateur, est de $0,65$. Ce score correct indique la cohérence des estimations visuelles de observateurs.

Pouvoir discriminant de la métrique. Nous avons utilisé un indicateur statistique (*Pearson Product Moment Correlation*) pour évaluer le pouvoir discriminant des métriques. Notre métrique 3D-NPRI dépasse largement les autres métriques. Elle permet donc une très bonne discrimination entre les segmentations en accord avec le jugement des observateurs humains.

Tous les détails concernant le pouvoir discriminant de notre métrique sont donnés dans la section 6.6 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 du présent document.

Nous avons également publié une étude encore plus détaillée de ces expériences subjectives, ainsi qu'une évaluation sujective des méthodes de segmentation automatique lors du *IEEE International Workshop on Multimedia Signal Processing (MMSP)* en 2010 (Benhabiles et al. 2010a).

4.5 Application à l'évaluation d'algorithmes de segmentation automatique

Nous avons utilisé notre nouvelle métrique, 3D-PRI et sa version normalisée 3D-NPRI, pour évaluer quelques méthodes connues de segmentation de maillages 3D. Nous avons également utilisé la collection de l'Université de Princeton (Chen et al. 2009).

4.5.1 Les méthodes de segmentations évaluées

Pour le benchmark de l'Université de Princeton (Chen et al. 2009), nous avons considéré les six méthodes récentes de segmentation automatique de maillages 3D utilisées justement dans la publication de Chen et al. (2009) car les résultats sont accessibles en ligne. Les six algorithmes sont les suivants : Attene et al. (2006a) (*fitting primitives*), Lai et al. (2008) (*random walk*), Golovinskiy et Funkhouser (2008) (deux algorithmes : *normalized cuts / randomized cuts*), Katz et al. (2005) (*core extraction*), et Shapira et al. (2008) (*shape diameter function*).

Pour notre benchmark, nous n'avons pu utiliser que deux algorithmes parmi les six précédents : Attene et al. (2006a) et Shapira et al. (2008). En effet, seuls ces deux algorithmes ont pu être utilisés pour produire les segmentations de notre collection vérité-terrain car leurs codes (source ou executable) sont disponibles en ligne.

4.5.2 Résultats

Les figures 4.4 et 4.5 montrent les scores 3D-NPRI calculés respectivement sur la collection de l'Université de Princeton et sur notre collection de maillages 3D et pour chaque algorithme cité précédemment. Ce type de graphe est utilisé par Unnikrishnan et al. (2007) pour l'évaluation de segmentations d'images 2D. Les valeurs sont données par ordre croissant pour chaque algorithme. Par conséquent, le $i^{ème}$ modèle n'est pas nécessairement le même pour chaque algorithme.

Le tableau 4.1 montre, pour chaque collection, le classement de chaque algorithme, selon le score 3D-NPRI, et la valeur moyenne de ce score.

Les figures 4.4 et 4.5 et le tableau 4.1 montrent bien que les six algorithmes sont pertinents puisque les valeurs moyennes de 3D-NPRI sont bien supérieures à zéro. L'algorithme *randomized cuts* donne d'ailleurs les

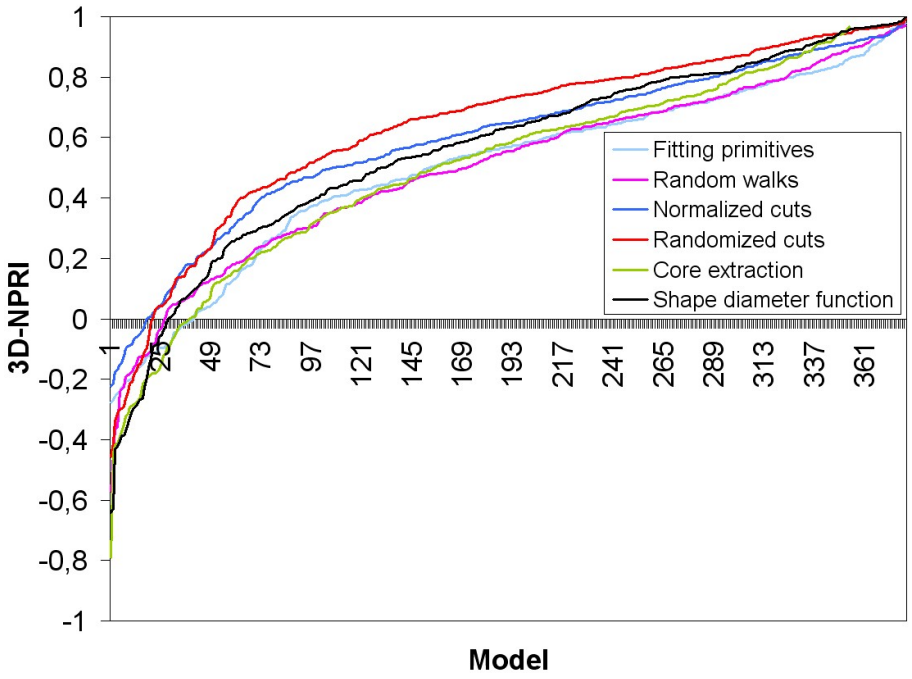


FIGURE 4.4 – *Scores 3D-NPRI classés par ordre croissant sur la collection de Chen et al. (2009).*

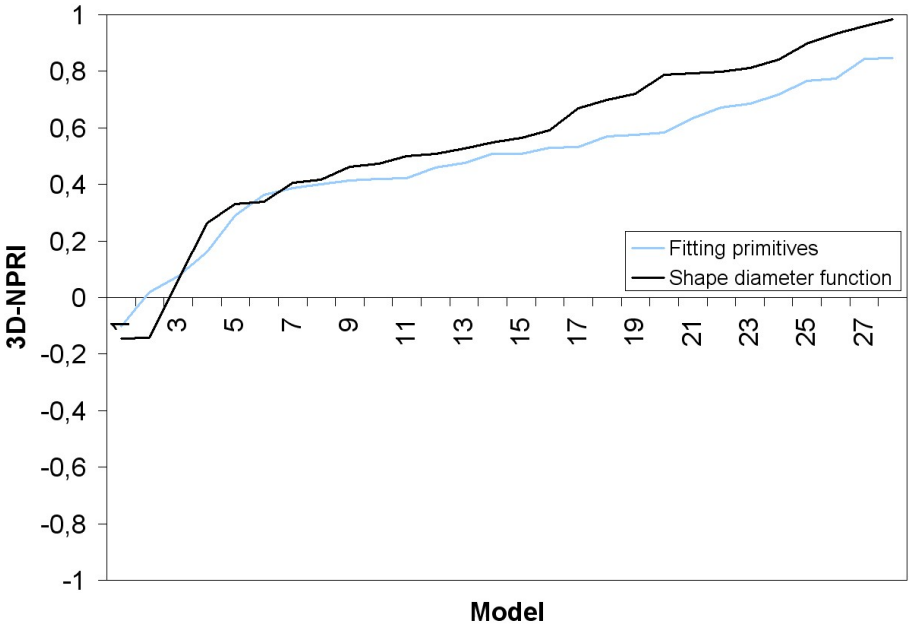


FIGURE 4.5 – *Scores 3D-NPRI classés par ordre croissant sur notre collection de maillages 3D.*

Algorithmes	3D-NPRI moyen	Rang
Fitting primitives	0.49/0.49	5/2
Random walks	0.50/-	4/-
Normalized cuts	0.59/-	2/-
Randomized cuts	0.63/-	1/-
Core extraction	0.46/-	6/-
Shape diameter function	0.56/0.55	3/1

TABLE 4.1 – *Classement de chaque algorithme sur la collection de Chen et al. (2009) et la nôtre.*

meilleurs résultats. Il est également intéressant de noter que les algorithmes *fitting primitives* et *shape diameter function* obtiennent des scores similaires dans les deux collections. Grâce à la qualité des segmentations vérités-terrains et malgré le nombre relativement peu élevé de modèles 3D, notre collection est très efficace pour l'évaluation d'algorithmes automatiques de segmentation.

Tous les détails concernant l'évaluation des méthodes de segmentation, ainsi qu'une étude sur les types de modèles 3D les mieux segmentés en fonction de l'algorithme utilisé, sont donnés dans la section 7 de la publication *A comparative study of existing metrics for 3D-mesh segmentation evaluation* (Benhabiles et al. 2010b) reproduite en annexe D page 191 du présent document.

4.5.3 Notre benchmark en ligne

Notre collection vérité-terrain et notre métrique 3D-PRI / 3D-NPRI forment un puissant outil de benchmark des méthodes de segmentation automatique de maillages 3D.

Nous avons mis ce benchmark à la disposition des chercheurs du domaine sous la forme d'un site web accessible à l'adresse `http://www-rech.telecom-lille1.eu/3dsegbenchmark/`. Ce site propose bien sûr le téléchargement de la collection vérité-terrain (figure 4.6) dans différents formats (VRML, PLY, OFF) mais également des outils simples à utiliser pour le chercheur qui souhaite évaluer sa méthode de segmentation de maillages 3D. Les résultats des comparaisons de différents algorithmes sont aussi visibles sous la forme de graphes et de moyennes de scores 3D-NPRI (figure 4.7) et sont exportables dans différents formats (PDF, CSV, etc.). Enfin, outre une documentation complète, une vidéo

FIGURE 4.6 – *Présentation et téléchargement de la collection vérité-terrain sur le site web.*

FIGURE 4.7 – *Visualisation des résultats de comparaison des méthodes de segmentation automatique présentes sur le site web.*

de démonstration est disponible [9] et donne un aperçu des possibilités du benchmark en ligne.

4.6 Segmentation par apprentissage

La segmentation de maillages 3D est très souvent un outil de base pour d'autres applications comme la reconnaissance de forme, la compression de maillages, le placage de textures, etc.

La plupart des méthodes actuelles analysent des critères de bas niveau ou des informations topologiques de la surface représentée par le maillage. Mais comme le soulignent Kalogerakis et al. (2010), le principal inconvénient de ces méthodes est leur limitation à un type de règle ou de caractéristique. Elles ne sont donc pas adaptées à une segmentation qui requerrait une combinaison de ces critères.

La création de collections de maillages 3D accompagnés de segmentations vérités-terrains comme celle de Chen et al. (2009) ou la nôtre – décrite précédemment dans ce chapitre – ouvre la voie vers l'apprentissage de la segmentation à partir de segmentations réalisées par des humains.

A notre connaissance, seuls les travaux de Kalogerakis et al. (2010) et les nôtres, existent dans ce domaine et on prouvé l'efficacité de l'apprentissage pour segmenter des maillages 3D.

Nous avons donc proposé une nouvelle méthode de segmentation de maillages 3D par apprentissage des arêtes frontières de régions (segments) dans une collection verité-terrain.

4.6.1 Notre algorithme de segmentation par apprentissage

Notre algorithme de segmentation par apprentissage se décompose en deux étapes principales : une étape *offline* d'apprentissage à partir d'une collection vérité-terrain, et une étape *online* de segmentation d'un maillage inconnu.

L'algorithme est entièrement détaillé dans la section 3 de la publication *Learning boundary edges for 3D-mesh segmentation* (Benhabiles et al. 2011) reproduite en annexe E page 208 de ce mémoire. Dans la suite de cette section, nous résumons les deux grandes étapes de notre méthode.

9. http://www-rech.telecom-lille1.eu/3dsegbenchmark/demo.html

Étape offline – apprentissage

Nous considérons l'apprentissage d'arêtes frontières de régions comme un problème de classification. La classification est apprise à partir de la collection vérité-terrain grâce au classifieur *AdaBoost* (Freund et Schapire 1995).

Le principe de base d'AdaBoost est de fournir, à partir d'un ensemble d'apprentissage, une fonction H de décision *frontière / non-frontière*. L'ensemble d'apprentissage est constitué de vecteurs calculés pour chaque arête de chaque maillage de la collection vérité-terrain. Un tel vecteur comporte 33 éléments calculés à partir de caractéristiques géométriques de l'arête associée : angle dièdre, différentes courbures et informations liées à la courbure – comme le *Shape Index* de Koenderink et van Doorn (1992) –, et la *Shape Diameter Function* de Shapira et al. (2008). Un label L et ensuite associé à chaque vecteur : $L = +1$ si l'arête en question est une arête frontière selon la vérité-terrain, $L = -1$ si l'arête ne l'est pas.

Finalement, à partir d'un vecteur caractéristique correspondant à une arête à tester, la fonction H apprise donne une valeur positive si l'arête en question est frontière de région ou une valeur négative si elle ne l'est pas.

Étape online – segmentation

La figure 4.8 montre des exemples de classification d'arêtes par la fonction H apprise. La ligne du haut de la figure montre une décision binaire pour chaque arête x : frontière (en rouge) si $H(x) > 0$, non-frontière sinon. La ligne du bas de la figure montre les valeurs de la fonction $H(x)$ selon l'échelle de couleurs indiquée sur la droite. Notons que, pour des raisons de qualité de l'illustration, nous avons coloré les sommets incidents des arêtes et non les arêtes elles-mêmes.

Ces résultats bruts – c'est-à-dire provenant directement de la fonction H – ne peuvent pas être utilisés directement sous cette forme pour segmenter les maillages.

Nous proposons un traitement du problème en plusieurs étapes illustrées à la figure 4.9.

Tout d'abord, il faut bien évidement calculer la valeur de la fonction H pour chaque arête du maillage 3D à segmenter (figure 4.9(a)). Toutes les arêtes dont la valeur de H est positive forment des régions d'intérêt (en rouge sur la figure 4.9(b)). Chaque région d'intérêt est un ensemble

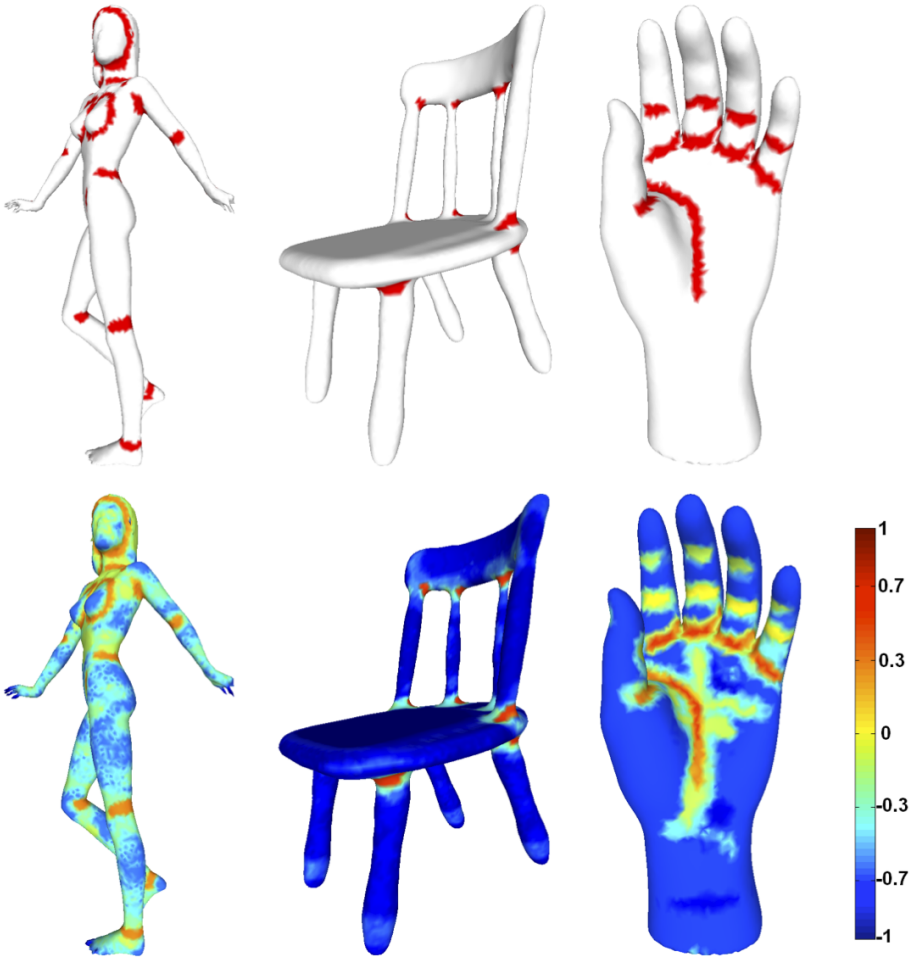
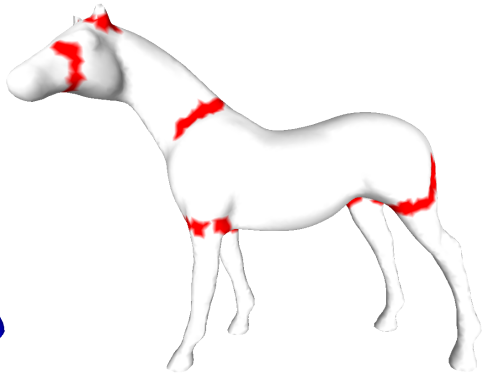
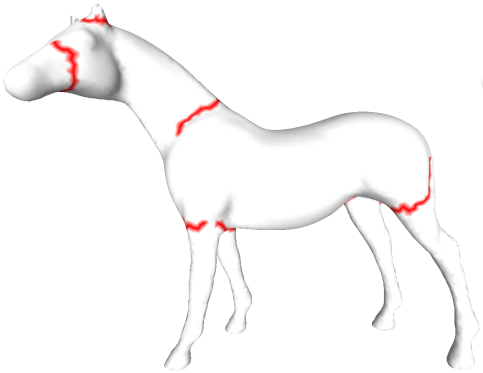


FIGURE 4.8 – *Classification par la fonction H d'arêtes sur trois modèles 3D. En haut : arêtes frontières en rouge. En bas : valeurs de la fonction H selon l'échelle de couleurs.*

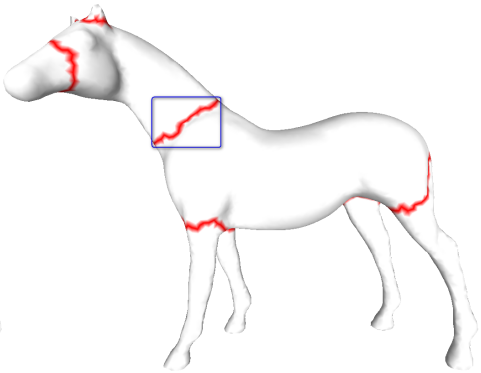
(a) *Calcul de la fonction H pour chaque arête.*



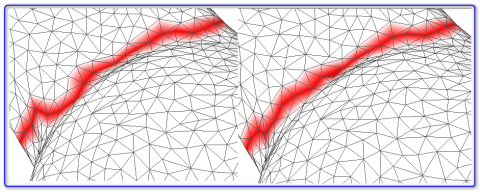
(b) *Extraction des régions d'intérêt.*



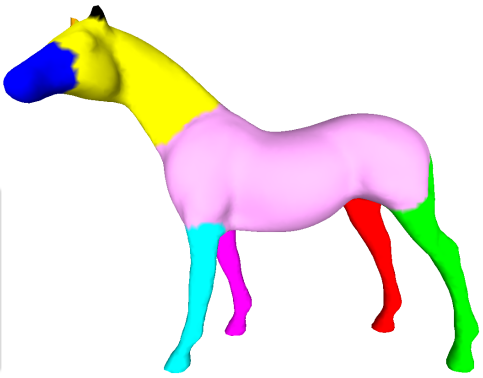
(c) *Amincissement des régions.*



(d) *Fermeture des contours.*



(e) Snake movement.



(f) *Segmentation finale.*

FIGURE 4.9 – *Aperçu de la chaîne d'obtention d'une segmentation finale.*

d'arêtes connectées entre elles. Puis un algorithme d'amincissement (Hubeli et Gross 2001) est appliqué à chaque région d'intérêt pour transformer chacune d'elles en un simple contour linéaire (figure 4.9(c)). Il reste ensuite à fermer ces contours grâce à une version améliorée de l'algorithme de Lee et al. (2005) basée sur la fonction H. Nous obtenons alors des contours fermés (figure 4.9(d)) qui donnent ainsi une première version des frontières de régions et donc une première version de la segmentation du maillage. Mais la plupart de ces frontières ne sont ni lisses ni réellement précises puisque l'étape d'amincissement ne prend en compte aucune information géométrique. Pour corriger ces défauts, nous leur appliquons une version adaptée de l'algorithme de *snake movement* (Jung et Kim 2004) basé à nouveau sur la fonction H. Cet algorithme améliore l'aspect et la qualité des frontières (figure 4.9(e)) dans le sens où celles-ci suivent à présent mieux la géométrie locale – crêtes et vallées – de la surface. Notons également que cet algorithme n'affecte pas la connectivité du maillage. Enfin, cet ensemble de frontières améliorées défini la segmentation final du maillage en entrée (figure 4.9(f)).

4.6.2 Expériences et résultats

Nous avons évalué notre méthode de segmentation grâce à différentes expériences. Nous avons également comparé notre méthode à d'autres méthodes de segmentation de l'état de l'art. Pour cela, nous avons utilisé deux benchmarks dédiés à l'évaluation de segmentations de maillages 3D : le benchmark de l'université de Princeton (Chen et al. 2009) et notre propre benchmark décrit plus haut dans ce chapitre.

Les évaluations et expériences sont détaillées dans la section 4 de la publication *Learning boundary edges for 3D-mesh segmentation* (Benhabiles et al. 2011) reproduite en annexe E page 208 de ce mémoire. Dans la suite de cette section, nous résumons les sept expériences que nous avons menées et donnons les principaux résultats.

Apprentissage par catégorie et apprentissage global

Pour cette première expérience, nous avons effectué l'apprentissage et les tests sur la collection de l'université de Princeton décrite à la section 4.3.3 page 94 de ce chapitre.

Nous avons testé deux stratégies d'apprentissage : l'apprentissage par catégorie et l'apprentissage global. L'apprentissage par catégorie consiste à

effectuer l'apprentissage et les tests catégorie d'objet par catégorie d'objet. Dans le premier cas, les performances de segmentation sont excellentes mais il est nécessaire de connaître la catégorie du modèle à segmenter *a priori*. Dans le second cas, les performances sont un peu moins bonnes bien évidemment, mais il n'est pas nécessaire de connaître la catégorie du modèle avant de le segmenter. Détails de cette expérience à la section 4.1 de l'article reproduit en annexe E de ce document.

La figure 4.10 montre quelques exemples de segmentations pour trois modèles de la collection de l'université de Princeton selon plusieurs algorithmes. De la gauche vers la droite apparaissent : la moyenne des segmentations vérités-terrains du benchmark de l'université de Princeton (Chen et al. 2009), la segmentation obtenue par notre algorithme par apprentissage (par catégorie), la segmentation obtenue par Kalogerakis et al. (2010), la segmentation obtenue par Golovinskiy et Funkhouser (2008), et la segmentation obtenue par Shapira et al. (2008). Nos segmentation sont clairement les meilleures, en termes de ressemblance aux vérités-terrains et de précision des frontières de segments.

Réutilisabilité de l'apprentissage

Pour cette deuxième expérience, nous avons voulu montré que l'apprentissage peut se faire sur une collection vérité-terrain et la segmentation sur des modèles qui ne sont pas dans cette dernière. Pour cela, nous avons effectué l'apprentissage sur la collection de l'université de Princeton et les tests sur des objets provenant de notre collection vérité-terrain.

Le tableau 4.2 montre les scores 3D-NPRI moyens calculés sur l'ensemble de notre collection pour différentes méthodes de segmentation. Notre méthode par apprentissage, même en apprentissage global, surpasse les autres.

La figure 4.11 montre des segmentations obtenues par notre algorithme de segmentation par apprentissage (global) sur des modèles provenant de diverses sources, autres que la collection de l'université de Princeton sur laquelle s'est effectué l'apprentissage.

Plus de détails sur cette expérience à la section 4.2 de l'article reproduit en annexe E de ce document.

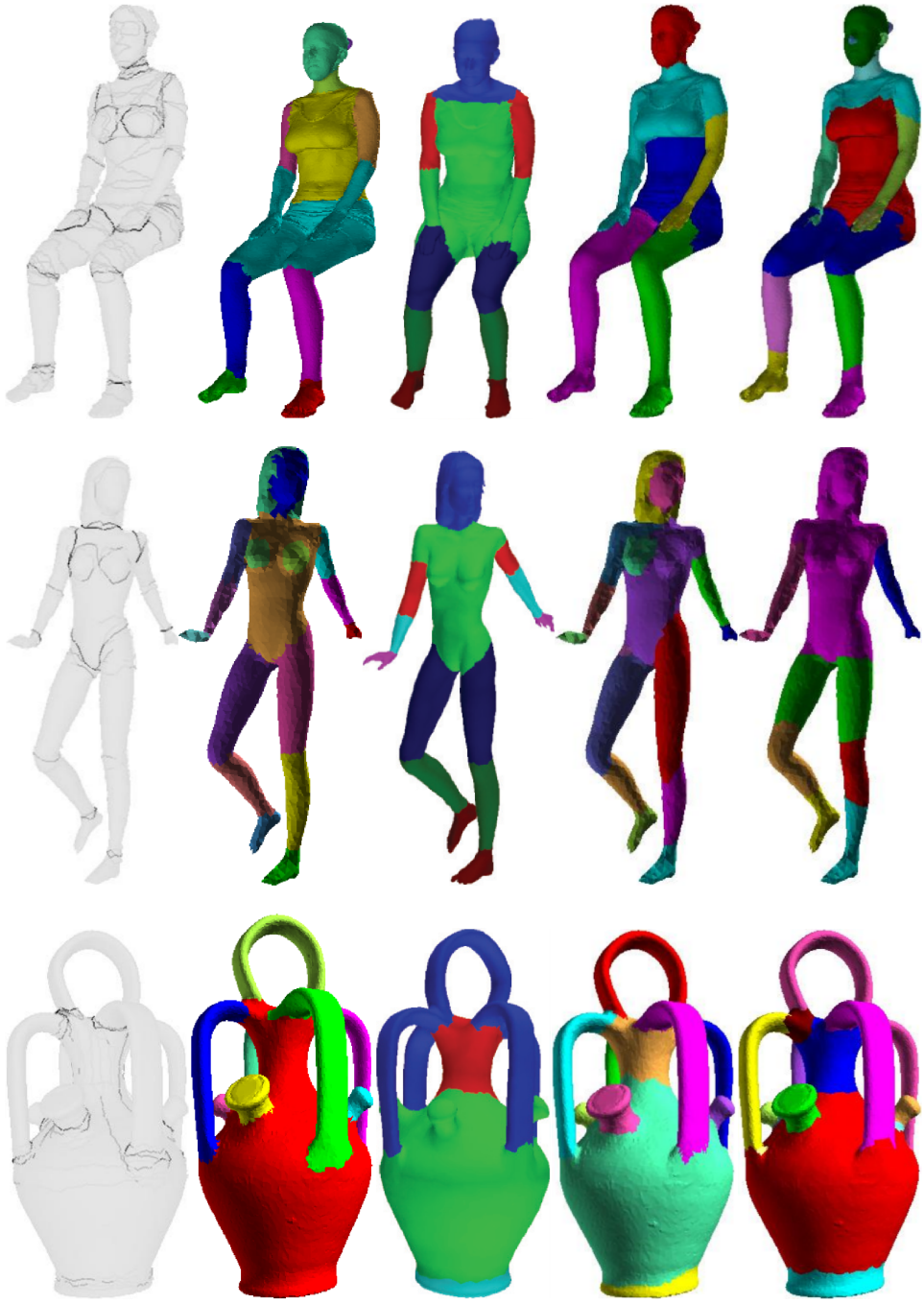


FIGURE 4.10 – *Exemples de segmentations automatiques de trois modèles par différentes méthodes. De la gauche vers la droite : moyenne des vérités-terrains du benchmark Princeton (Chen et al. 2009), notre algorithme par apprentissage, algorithmes de Golovinskiy et Funkhouser (2008), de Golovinskiy et Funkhouser (2008), et de Shapira et al. (2008).*

Algorithmes	3D-NPRI moyen
Notre méthode (apprentissage global)	0.60
Shape diameter function (Shapira et al. 2008)	0.55
Fitting primitives (Attene et al. 2006a)	0.49
Reeb graphs (Tierny et al. 2007b)	0.45
CAD-curvatures (Lavoué et al. 2005)	0.34

TABLE 4.2 – *Scores 3D-NPRI moyens de plusieurs méthodes testées sur notre collection vérité-terrain. Notre méthode a réalisé son apprentissage (global) sur la collection de l'université de Princeton.*

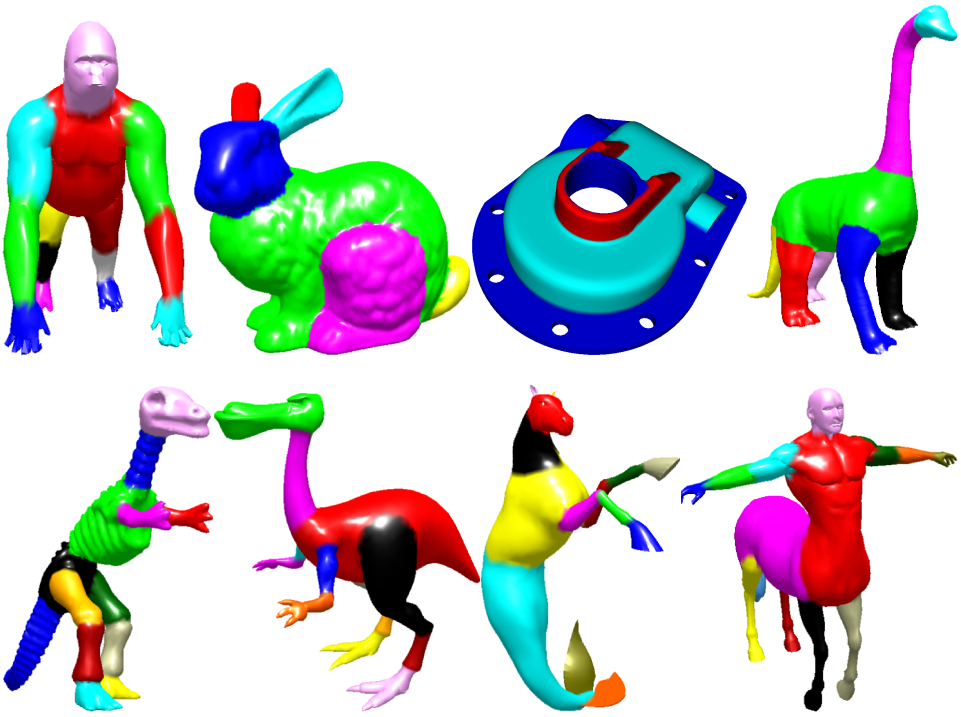


FIGURE 4.11 – *Segmentations obtenues par notre algorithme d'apprentissage (global sur la collection de l'université de Princeton) pour plusieurs modèles 3D provenant de diverses sources.*

Efficacité selon la catégorie de l'objet à segmenter

Dans cette troisième expérience, un apprentissage par catégorie a été effectué pour se rendre compte si une ou plusieurs catégories étaient particulièrement bien – ou mal – traitées par notre algorithme.

Les scores 3D-NPRI moyens pour chaque catégorie, sur chacune des deux collections vérités-terrains, étant très bons (compris entre 0.5 pour la catégorie *bustes*, toujours très difficile à traiter, et 0.95), toutes les catégories sont bien traitées par notre algorithme par apprentissage. Détails de cette expérience à la section 4.3 de l'article reproduit en annexe E de ce document.

Influence du *snake movement* sur la qualité des segmentations

La quatrième expérience étudie l'influence du *snake movement* (Jung et Kim 2004) sur l'amélioration de la qualité des frontières de segments et donc de la segmentation. Cette expérience (voir les détails en section 4.4 de la publication reproduite en annexe E de ce mémoire) montre l'amélioration sensible de la qualité des segmentations obtenues en utilisant le *snake movement*, quelque soit la catégorie du modèle segmenté.

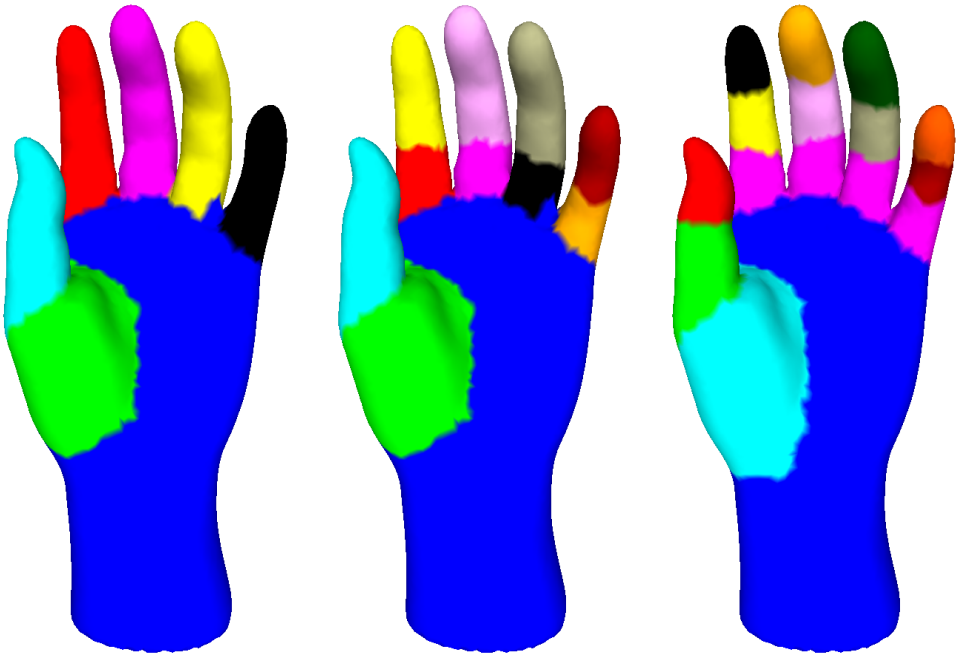
Interaction avec l'utilisateur

Notre méthode de segmentation par apprentissage est entièrement automatique et ne dépend d'aucun paramètre extérieur. On a cependant imaginé, dans cette cinquième expérience, de donner la possibilité à l'utilisateur d'agir sur certains paramètres comme le seuil appliqué la fonction apprise H ou d'autres choses. Détails à la section 4.5 de l'article reproduit en annexe E du présent document.

La figure 4.12 montre l'influence de ce seuil dans l'obtention de différents niveaux de finesse dans la segmentation d'un objet 3D de la catégorie *mains*.

Robustesse à certaines transformations

Dans cette sixième expérience, nous nous intéressons à la robustesse de notre algorithme par apprentissage selon le change de la posture (pose) d'un modèle, et selon le bruit présent sur le maillage.



(a) $H(x) > 0.3$ (b) $H(x) > 0$ (c) $H(x) > -0.2$

FIGURE 4.12 – *Trois niveau de finesse de segmentation selon le seuil appliqué à la fonction H.*

Le plus intéressant ici étant le changement de posture, la figure 4.13 montre les segmentations obtenues pour le modèle *armadillo* et un modèle de la catégorie *humains* dans différentes postures. Les segmentations sont globalement stables, ce qui montre une bonne robustesse de notre méthode au changement de pose.

L'influence du bruit sur le maillage est détaillé à la section 4.6 de l'article reproduit en annexe E de ce document.

Temps d'execution

La septième expérience n'en est pas une à proprement parlé, mais elle donne un ordre de grandeur du temps d'exécution de notre algorithme par apprentissage.

Les expériences décrites ci-dessus ont été réalisées sur un PC équipé d'un processeur Intel Core 2 Duo cadencé à 3 GHz et de 4 Go de mémoire vive. Globalement, l'étape d'apprentissage *offline* se déroule en moins de dix minutes pour toutes les catégorie, exceptée la catégorie *armadillo* pour laquelle il faut environ 40 minutes. En effet, la taille des mailages de la catégorie *armadillo* est plus importante que les autres. Concernant l'étape

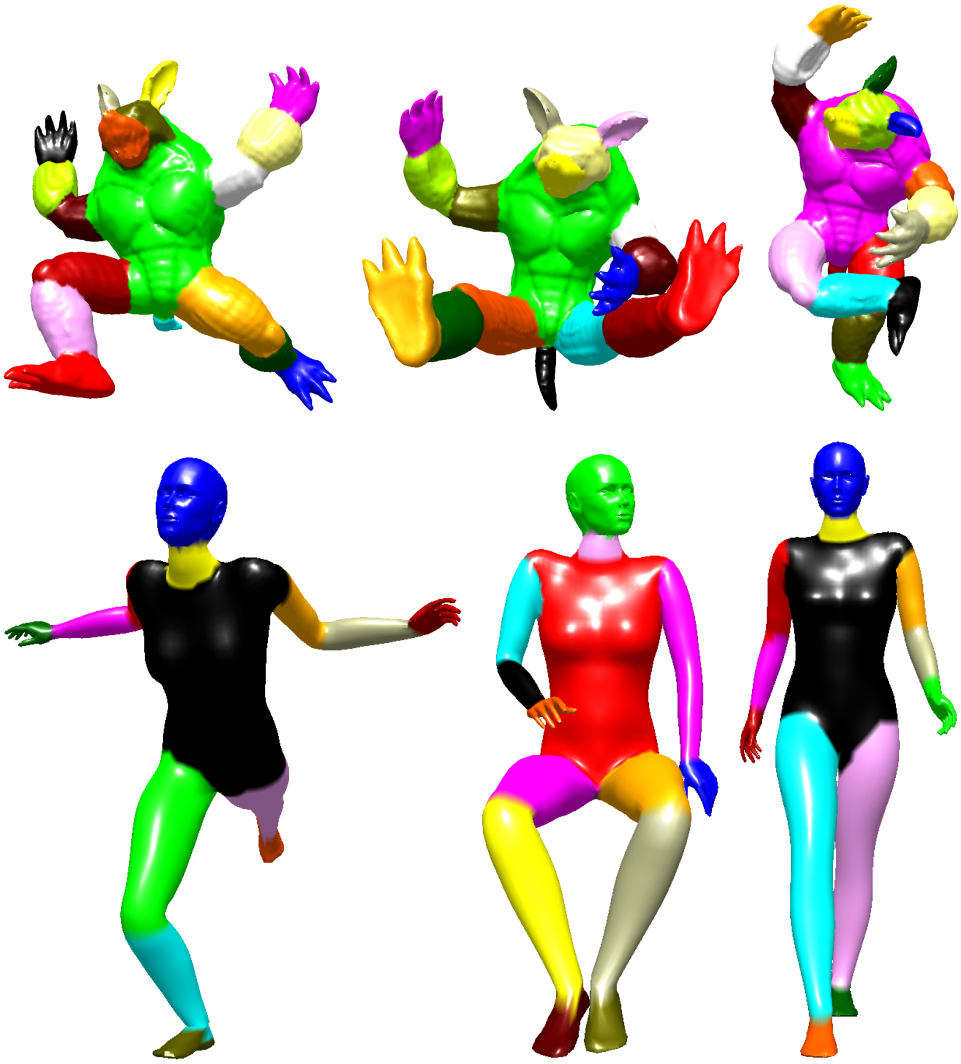


FIGURE 4.13 – *Robustesse de notre algorithme par apprentissage au changement de pose.*

de segmentation *online*, on peut dire qu'elle se déroule en temps presque interactif, environ 1 minute par modèle, sauf, une fois de plus et pour les mêmes raisons, pour la catégorie *armadillo*.

4.7 Conclusion du chapitre

Nous avons présenté dans ce chapitre les travaux de recherche menés en segmentations de maillages 3D avec l'encadrement de la thèse d'Halim Benhabiles. Les deux objectifs de ces recherches étaient d'une part l'évaluation des méthodes de segmentation automatiques de maillages 3D, et d'autre part la mise au point d'une nouvelle méthode de segmentation qui

prenne en compte des connaissances *a priori*, apportées par la compréhension humaine, sur des objets courants.

Le premier objectif de ces travaux répond à un besoin évident d'évaluer de manière quantitative et le plus objectivement possible la qualité des segmentations produites par un algorithme. Jusque là en effet, les contributeurs de ce domaine de recherche ne proposaient généralement qu'une appréciation visuelle et subjective de leurs résultats. Certains, comme Attene et al. (2006b), se sont néanmoins intéressés à des critères objectifs et évaluables, mais sans prendre en compte de connaissances *a priori* sur les segmentations issues de la connaissance humaine. Nous avons donc proposé le premier benchmark d'évaluation des méthodes de segmentation automatique de maillages 3D, qui utilise une collection de segmentations réalisées par des volontaires et appelée *collection vérité-terrain*, et une métrique performante et répond à de nombreux critères de qualité pour comparer une segmentation automatique à de multiples segmentations vérités-terrains. Nous répondons en cela à la première question soulevée à la fin du chapitre précédent.

Le second objectif poursuit l'effort mené dans la prise en compte de la connaissance humaine. Nous avons mis au point une nouvelle méthode de segmentation automatique de maillages 3D qui utilise cette connaissance à travers la collection de segmentations vérités-terrains créée précédemment. Notre méthode utilise des techniques d'apprentissage. Nous avons prouvé que notre méthode produit des segmentations de qualité, aussi bien en terme de similarité aux segmentations vérités-terrains, qu'en terme de précision et de lissage des frontières entre segments. Nous avons donc apporté une réponse (parmi d'autres, sans aucun doute) à la deuxième question soulevée à la fin du chapitre précédent.

Pour autant, nos solutions restent perfectibles. Bien que moins précise dans la définition des frontières de segments, la collection vérité-terrain de l'université de Princeton (Chen et al. 2009) offre à la communauté des chercheurs du domaine un plus grand nombre de modèles 3D. La méthode de segmentation par apprentissage proposée par Kalogerakis et al. (2010) est performante et offre, en plus de cela, l'étiquetage (*labeling*) automatique des différentes régions d'une segmentation. Enfin, dans notre méthode par apprentissage, la fonction apprise H n'utilise que des caractéristiques géométriques existantes et n'innove pas de ce côté-là. L'ajout de nouvelles caractéristiques améliorera très certainement encore notre méthode.

Conclusion et perspectives

Au cours de ce mémoire, j'ai présenté mes activités de recherche et les contributions majeures apportées depuis ma thèse de doctorat et ma nomination en tant que Maître de Conférences de l'Institut Télécom en décembre 2002.

Ce mémoire apporte des éléments de réponse aux deux questions qui guident mes recherche sur les modèles 3D depuis le début : « Comment retrouver un modèle 3D similaire à celui ou ceux en ma possession dans une collection de modèles 3D ? » et « Comment retrouver la structure d'un modèle 3D maillé sans connaissance *a priori* sur celui-ci ? » Les réponses apportées à cette dernière question en ont ensuite soulevé deux autres concernant la segmentation de maillages 3D auxquelles des réponses ont été apportées dans le dernier chapitre.

5.8 Résumé et principales contributions

Pour répondre à la question « Comment retrouver un modèle 3D dans une collection de modèles ? », nous avons proposé une approche du type *recherche par l'exemple*. La première contribution de cette partie du mémoire concerne la représentation intuitive d'un objet 3D par un ensemble de vues 2D qui le caractérise. Nous avons appelé ces vues 2D *vues caractéristiques*. La comparaison de deux objets 3D est donc simple et intuitive : deux objets 3D sont visuellement similaires si les vues qui les caractérisent le sont aussi. La seconde contribution concerne la façon d'utiliser ces vues caractéristiques pour indexer et retrouver les modèles 3D. Nous prenons en compte le fait que certaines vues ont plus d'importance que d'autres dans la caractérisation de l'objet et intégrons des probabilités dans la recherche des modèles 3D.

La deuxième question a ensuite été abordée et nous y avons répondu en nous intéressant à l'étude de la topologie des maillages 3D – principal, mais pas unique, moyen de représenter les modèles 3D. Grâce à

cela, nous avons mis au point un algorithme rapide et robuste d'extraction des points caractéristiques d'un maillage 3D. Nous avons ensuite proposé une méthode d'extraction de squelette significatif d'un maillage 3D. Enfin, l'étude des constrictions sur les composantes proéminentes nous a permis d'améliorer encore la qualité du squelette extrait.

Nous avons ensuite testé différentes applications de ce squelette extrait d'un maillage 3D : la déformation de maillage, l'indexation, et surtout la segmentation. Avec cette dernière application sont apparues deux autres questions : « Comment évaluer le plus objectivement possible la qualité des segmentations obtenues ? » et « Peut-on exploiter la connaissance humaine sur les objets pour segmenter des maillages 3D ? »

Nous avons apporté, dans le dernier chapitre, des éléments de réponse à ces deux dernières questions. Nous avons proposé un benchmark pour l'évaluation des méthodes de segmentation automatique. Ce benchmark est composé d'une collection de maillages 3D avec plusieurs segmentations vérités-terrains réalisées par des volontaires humains, ainsi qu'une métrique de comparaison de segmentation répondant à un ensemble de critères de qualité. Enfin, pour prendre en compte la connaissance humaine sur les objets, nous avons développé un nouvel algorithme de segmentation de maillages 3D qui "apprend" à partir des segmentations vérités-terrains provenant de notre collection ou d'une autre.

5.9 Perspectives

Je livre ici les questions, et éléments de réflexion pour la suite de mes recherches à moyen et long terme. Certaines des perspectives énoncées dans la suite sont d'ailleurs engagées à ce jour.

5.9.1 Indexation invariante à la pose et par partie

Notre approche de l'indexation de modèles 3D par vues caractéristiques considère tous les modèles de la base de données comme des objets rigides. Or, comme je l'ai déjà souligné dans les chapitres précédents, la majeure partie des objets peut se présenter sous différentes postures ou poses. Ce sont donc des objets non-rigides. Ces changements de pose sont le résultat de transformations sur certaines parties des objets et induit également souvent un changement dans la topologie du maillage. La figure

5.1 montre quelques exemples, extraits du projet TOSCA [10], d'un modèle 3D d'humanoïde avec des changements de topologie (indiqués par des cercles rouges).

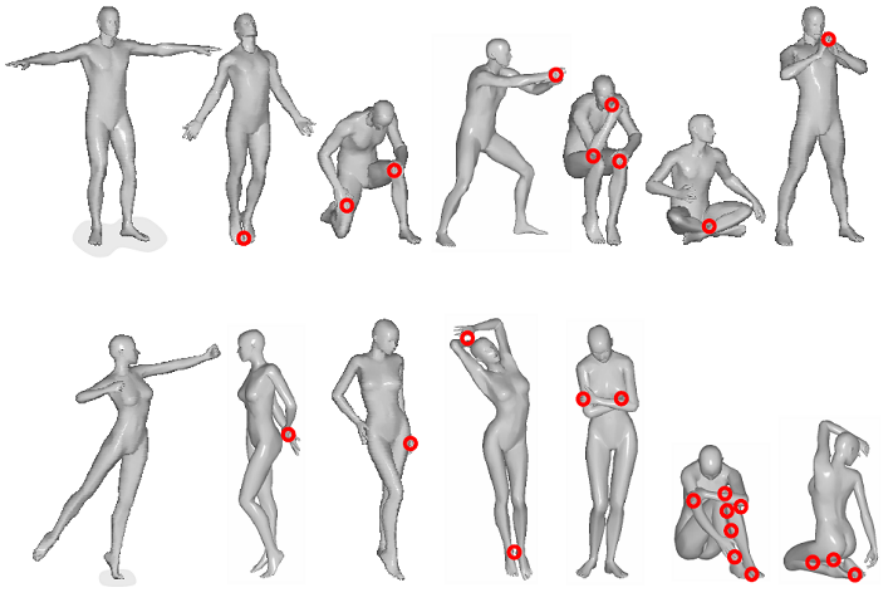


FIGURE 5.1 – *Un modèle 3D d'humanoïde dans différentes poses induisant souvent des changements de topologie (cercles rouges). Ces exemples sont tirés de la base TOSCA.*

Nous avons fait une première proposition pour l'indexation de modèles 3D par parties par l'intermédiaire des squelettes extraits des maillages – voir la section 3.3.2 page 78 du chapitre 3 de ce mémoire.

La thèse d'Hedi Tabia, soutenue le 27 septembre 2011 portait en partie sur ce sujet. Nous avons utilisé un descripteur basé sur l'utilisation de courbes centrées sur des points caractéristiques pour indexer les modèles 3D et sur les fonctions de croyance pour effectuer la recherche. Ces travaux sont détaillés dans la publication *A New 3D-Matching Method of Nonrigid and Partially Similar Models Using Curve Analysis* (Tabia et al. 2011b) reproduite en annexe E page 208 de ce document. Nous avons également proposé une méthode (Tabia et al. 2011a;c) qui utilise l'approche des *Bags of Features*, technique dérivée des *Bags of Words* utilisée pour la recherche dans les grandes bases de documents textes.

La thèse de Rachid El Khoury, commencée en novembre 2009 et financée par le programme *Futur et Ruptures* de l'Institut Télécom, porte également sur l'indexation par partie. Dans cette thèse, l'attention est plus particulièrement portée sur l'utilisation d'une signature de forme de type *diffusion de la chaleur* (Sun et al. 2009). Pour l'instant, nous avons proposé

10. http://tosca.cs.technion.ac.il/

une nouvelle fonction d'application pour le calcul du graphe de Reeb d'un maillage 3D. Cette fonction porte sur la distance *commute-time* et la distance de diffusion (El Khoury et al. 2012). Nous utilisons ensuite les courbes de niveau produites par cette fonction pour les utiliser à travers les *Bags of Features* afin d'indexer une collection de maillages 3D.

Quelques applications

L'indexation invariante à la pose et par parties ouvre de nombreuses possibilités d'applications. Parmi celles-ci, on peut citer l'utilisation de requêtes partielles pour l'interrogation des moteurs de recherche de modèles 3D, ainsi que la modélisation par l'exemple.

Requêtes partielles. Est-il toujours justifié d'exiger que la requête-exemple émise par l'utilisateur représente toujours l'objet recherché en entier ? Par conséquent, l'indexation de modèles 3D entiers est-elle suffisante ? La question de l'expression de requêtes partielles se pose donc. On peut en effet facilement imaginer qu'un utilisateur présente un objet *main* en requête et reçoive en réponse des modèles 3D représentant une main mais aussi des modèles 3D plus complexes dont la main n'est qu'une partie. De plus, certains objets sont des créations composites. Par exemple, la figure 5.2 montre un modèle 3D de centaure, créature mythologique mi-homme mi-cheval. La comparaison de l'objet composite avec ses objets de base est un problème qui se pose depuis peu de temps (Bronstein et al. 2008).

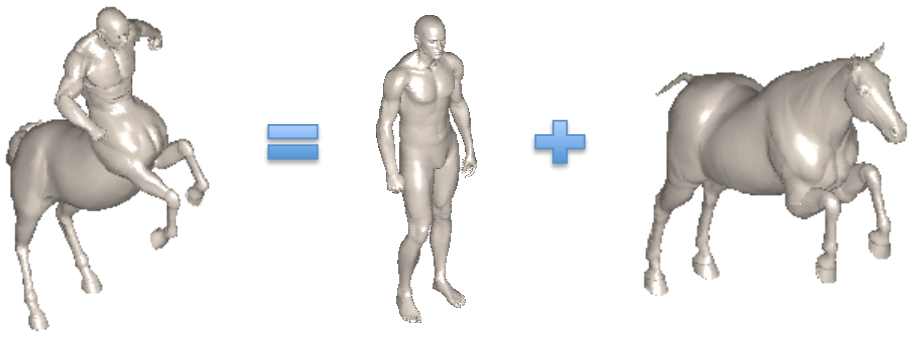
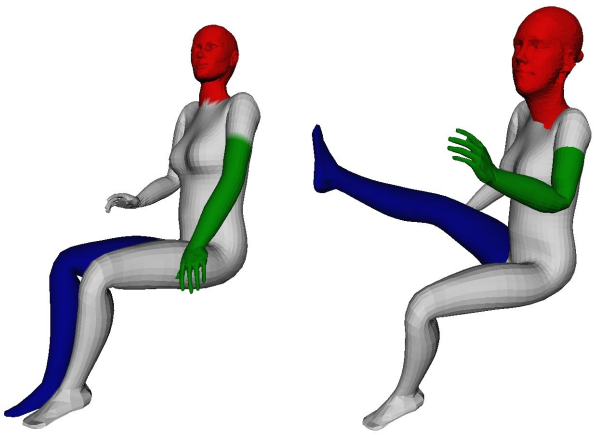
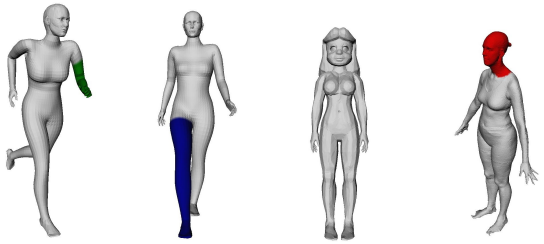


FIGURE 5.2 – *Objet composite : le centaure est une créature mi-homme mi-cheval.*

Modélisation par l'exemple. La création de nouveaux contenus tridimensionnels est un travail long, complexe et coûteux. Réutiliser les objets 3D qui existent déjà dans les bases de données des produc-



(a) *Objet-requête.* (b) *Nouveau modèle 3D.*



(c) *Modèles retournés par le moteur de recherche.*

FIGURE 5.3 – *Création d'un nouveau modèle 3D à partir de plusieurs modèles existant. Les parties échangées portent les mêmes couleurs.*

teurs de contenu, peut grandement accélérer et simplifier ce travail. Suivant l'exemple de Funkhouser et al. (2004), pionniers dans le domaine, nous avons déjà proposé une première ébauche de la modélisation par l'exemple en tant qu'application de notre méthode d'extraction de squelettes et d'indexation par partie. Cette proposition est décrite à la section 6 de la publication *Partial 3D shape retrieval by Reeb pattern unfolding* (Tierny et al. 2009), reproduite en annexe C page 175 de ce mémoire. La figure 5.3 montre un exemple de création d'un nouveau modèle 3D à partir de la modification d'un modèle existant, après requête dans une base de modèles 3D et échanges de parties entre l'objet-requête et les modèles-réponses retournés par le moteur de recherche. Les parties échangées portent les mêmes couleurs sur la figure.

Avec les méthodes d'indexation et de segmentation que nous avons développés, je souhaite aller beaucoup plus loin et proposer des outils beaucoup plus interactifs, à l'image de ce qu'ont proposé Lee et al. (2011) pour le guidage de dessin à main levé, transposé à la 3D.

5.9.2 Modèles 3D dynamiques (3D+t)

Ce mémoire s'intéresse exclusivement aux modèles 3D dits *statiques*, c'est-à-dire des maillages qui définissent la surface d'objets fixes, sans aucun mouvement. Or, de la même façon que la vidéo donne une illusion de mouvement grâce à une succession d'images fixes, il existe des maillages animés ou *dynamiques* qui sont une succession de maillages liés dans le temps et qui, visualisés rapidement, donnent un objet 3D en mouvement. Ces maillages sont d'ailleurs également appelés *maillages 3D+t* car la composante du temps t est prise en compte. La figure 5.4 montre trois étapes – donc trois maillages statiques – d'un maillage dynamique représentant un cheval au galop.

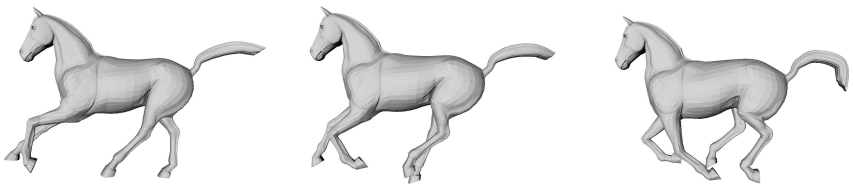


FIGURE 5.4 – *Trois étapes d'un maillage dynamique représentant un cheval au galop.*

Je suis persuadé que la modélisation et l'analyse de maillages évoluant dans le temps est l'un des grands challenges de mon domaine de recherche. Ces maillages sont en effet à la base de la télévision en trois dimensions *véritables*. En effet, depuis 2009 et la sortie du film *Avatar* de James Cameron, le terme 3D utilisé pour le cinéma – et depuis moins longtemps pour la télévision – est simplement un terme marketing pour parler de cinéma en relief. La révolution du spectacle cinématographique ou télévisuel en trois dimensions se fera quand le spectateur pourra suivre l'action (film, spectacle, rencontre sportive, etc.) du point de vue qu'il aura choisi. Une véritable modélisation tridimensionnelle dynamique sera alors indispensable.

Nous avons déjà proposé deux méthodes de segmentation de maillages dynamiques : l'une grâce aux squelettes topologiques et à l'étude de l'évolution des mouvements du maillage d'un instant au suivant (Tierny et al. 2008b) ; l'autre à l'aide de la fonction H de notre méthode de segmentation par apprentissage sur la série de maillages statiques constituant le maillage dynamique global (méthode non encore publiée). Cependant, ces méthodes ne fonctionnent que si les maillages conservent leurs connectivités, leur nombres de points et de facettes, etc. au cours du temps. Or,

comme l'ont montré les travaux du projet ANR MADRAS, les maillages dynamiques ne respectent pas tous, loin s'en faut, ces propriétés idéales.

5.9.3 Traitement de maillages *bruts* de scanners 3D

A l'instar des scanners d'images (2D) il y a quelques dizaines d'années, l'utilisation des scanners 3D se développe. Même le grand public commence à avoir accès à ces nouvelles technologies à travers le périphérique *Kinect* [11] de Microsoft pour sa console de jeux vidéos XBox, et bientôt pour les ordinateurs personnels. Mais là où un scan 2D donne des résultats satisfaisants très rapidement et avec peu de manipulations post-scan, les résultats obtenus par les scanners 3D, toutes technologies confondues, ne sont pas exploitables sans un post-traitement relativement lourd.

Les figures 5.5 et 5.6 montrent une session de scan avec un scanner laser Konica-Minolta VIVID910 [12] couplé avec un plateau tournant, d'un jouet *Woody*, personnage du film d'animation *Toy Story* (®Disney). Cette session de scan est le travail de mon collègue Boulbaba Ben Amor avec le matériel que nous possédons dans l'équipe MIIRE. La figure 5.5 montre quatre exemples de maillages partiels sur les quatorze obtenus dans la session de scan, pour obtenir l'objet 3D final représenté à la figure 5.6 avec ses textures.

Chaque scan partiel apporte de l'information pour former le maillage final. Cependant, en fonction de l'objet, de sa taille, de ses concavités, de la réponse de la matière face au laser, etc. les informations de chaque scan ne sont pas complètes et, même une fois reconstitué, on remarque que certaines parties de l'objet n'ont pu être scannées de manière satisfaisante : sur la figure 5.6, on voit bien que le bord inférieur du chapeau ainsi que les jambes par exemples, n'ont pas été obtenus. De plus, pour obtenir le résultat final, des traitements post-scans, manuels ou assistés par un opérateur, ont été nécessaires : bouchage de trous, élimination de parties du "décor", lissage de certaines parties, etc.

Malgré le caractère parcellaire de chaque scan, je pense qu'il est important de s'intéresser à des méthodes d'extraction automatique d'informations de ces scans. La segmentation et le calcul de descripteurs de formes adaptés seront une avancée considérable dans la compréhension automa-

11. http://www.xbox.com/kinect/

12. http://www.konicaminolta.com/instruments/products/3d/non-contact/vivid910/

tique des maillages 3D provenant de matériels de captures ou présentant des défauts de maillage.

FIGURE 5.5 – *Quatre maillages partiels parmi un total de quatorze obtenus par un scanner Konica-Minolta VIVID910 et son plateau tournant, d'un jouet* Woody, *personnage de* Toy Story *(®Disney). Chaque maillage a une résolution d'environ* 50 000 *points et* 96 000 *faces.*

FIGURE 5.6 – *Quatorze scans du personnage* Woody *de la figure précédente assemblés automatiquement grâce au logiciel Konica-Minolta* Polygon Editing Tool *fourni avec le scanner. Le maillage final a une résolution de* 162 196 *points et* 253 470 *faces.*

Bibliographie

Hirotugu Akaike. Information theory and extention of the maximum likehood principle. Dans *IEEE International Symposium on Information Theory*, pages 267–281, 1973. (Cité page 45.)

Mihael Ankerst, Gabi Kastenmuller, Hans-Peter Kriegel, et Thomas Seidl. Nearest neighbor classification in 3D protein databases. Dans *International Conference on Intelligent Systems for Molecular Biology*, 1999. (Cité page 49.)

Marco Attene, Silvia Biasotti, et Michela Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, Volume 19 :127–138, 2003. (Cité page 68.)

Marco Attene, Bianca Falcidieno, et Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer - International Journal of Computer Graphics*, 22(3) :181–193, 2006a. (Cité pages 102, 104 et 115.)

Marco Attene, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, et Ayellet Tal. Mesh segmentation, a comparative study. *IEEE International Conference on Shape Modeling and Applications*, pages 14–25, 2006b. (Cité pages 87, 90 et 119.)

A. Baskurt, F. Blum, M. Daoudi, J.L. Dugelay, F. Dupont, A. Dutartre, T. Filali Ansary, F. Fratani, E. Garcia, G. Lavoué, D. Lichau, F. Prêteux, J. Ricard, B. Savage, J.P. Vandeborre, et T. Zaharia. SEMANTIC-3D : compression, indexation et tatouage de données 3D. Dans *Compression et Représentation des Signaux Audiovisuels (CORESA'2004)*, Lille, France, 25-26 mai 2004. (Cité page 16.)

Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, et Mohamed Daoudi. A subjective experiment for 3D-mesh segmentation evaluation. Dans *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, Saint-Malo, France, October 4-6 2010a. (Cité page 103.)

Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, et Mohamed Daoudi. Learning boundary edges for 3D-mesh segmentation. *Computer Graphics Forum - Eurographics Association*, 2011. Accepted for publication on 28 April 2011, early view online (before inclusion in an issue) 13 June 2011. (Cité pages 88, 108 et 112.)

Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, et Mohamed Daoudi. A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. Dans *IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2009)*, Beijing, China, June 26-28 2009. (Cité pages 88, 90, 94 et 98.)

Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, et Mohamed Daoudi. A comparative study of existing metrics for 3D-mesh segmentation evaluation. *The Visual Computer - International Journal of Computer Graphics*, 26(12) :1451–1466, December 2010b. (Cité pages 97, 100, 101, 103 et 106.)

James O. Berger et Luis R. Pericchi. The intrinsic bayes factor for model selection and prediction. *The Journal of the American Statistical Association*, pages 109–122, 1994. (Cité page 45.)

Stefano Berretti, Alberto Del Bimbo, et Pietro Pala. 3d mesh decomposition using reeb graphs. *Image and Vision Computing*, 27(10) :1540–1554, September 2009. (Cité pages 90 et 97.)

Silvia Biasotti, Simone Marini, Michela Mortara, et Giuseppe Patanè. An overview on properties and efficacy of topological skeletons in shape modelling. Dans *Shape Modeling International*, pages 245–254, 2003. (Cité pages 68 et 77.)

Irving Biederman. Recognition-by-components : A theory of human image understanding. *Psychological Review*, 94 :115–147, 1987. (Cité pages 82 et 87.)

Jules Bloomenthal, éditeur. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., 1997. (Cité page 63.)

Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, et Bruno Levy. *Polygon Mesh Processing*. A K Peters, 2010. (Cité page 64.)

Alexander M. Bronstein, Michael M. Bronstein, Alfred M. Bruckstein, et Ron Kimmel. Partial similarity of objects, or how to compare a centaur

to a horse. *International Journal of Computer Vision*, 84(2), 2008. (Cité page 124.)

John F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986. (Cité page 55.)

Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, et Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3) : 223–232, September 2003. (Cité page 49.)

Xiaobai Chen, Aleksey Golovinskiy, et Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009. (Cité pages xii, xiii, 88, 90, 94, 95, 97, 98, 104, 105, 106, 108, 112, 113, 114 et 119.)

Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, et Valerio Pascucci. Loops in Reeb graphs of 2-manifolds. Dans *Symposium on Computational Geometry*, pages 344–350, 2003. (Cité pages 67, 69 et 71.)

Sébastien Delest, Romuald Boné, et Hubert Cardot. Réflexions sur la mise en place d'une vérité terrain pour la segmentation de maillages polygonaux 3d en parties significatives. Dans *Groupe de Travail en Modélisation Géométrique*, pages 49–62, mars 2008. (Cité page 91.)

Richard C. Dubes. How many clusters are best ? – an experiment. *Pattern Recognition*, 20(6) :645–663, 1987. (Cité page 44.)

Richard O. Duda et Peter E. Hart. *Pattern Classification and Scene Analysis*. Wiley-Interscience Publication, 1973. (Cité page 44.)

Jean-Luc Dugelay, Atilla Baskurt, et Mohamed Daoudi, éditeurs. *3D Object Processing : Indexing, Compression and Watermarking*. John Wiley & Sons, April 2008. (Cité pages 16 et 36.)

Rachid El Khoury, Jean-Philippe Vandeborre, et Mohamed Daoudi. 3d mesh reeb graph computation using commute-time and diffusion distances. Dans *SPIE Electronic Imaging 2012 Symposium (3D Image Processing (3DIP) and Applications)*, volume 8290, Burlingame, California, USA, January 22-26 2012. (Cité page 124.)

Tarik Filali Ansary, Mohamed Daoudi, et Jean-Philippe Vandeborre. A bayesian 3D search engine using adaptive views clustering. *IEEE Tran-*

sactions on Multimedia, 9(1) :78–88, January 2007. (Cité pages 45, 47, 49, 50 et 54.)

Tarik Filali Ansary, Jean-Philippe Vandeborre, et Mohamed Daoudi. A bayesian approach for 3D models retrieval based on characteristic views. Dans *17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, Cambridge, United Kingdom, August 23-26 2004a. (Cité page 38.)

Tarik Filali Ansary, Jean-Philippe Vandeborre, et Mohamed Daoudi. A bayesian framework for 3D models retrieval based on characteristic views. Dans *2nd IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'04)*, Thessaloniki, Greece, September 6-9 2004b. (Cité page 39.)

Tarik Filali Ansary, Jean-Philippe Vandeborre, et Mohamed Daoudi. A framework for 3D CAD models retrieval from 2D images. *Annals of Telecommunications, special issue on Technologies and tools for 3D imaging*, 60(11-12) :1337–1359, November-December 2005. (Cité page 44.)

Tarik Filali Ansary, Jean-Philippe Vandeborre, et Mohamed Daoudi. On 3D retrieval from photos. Dans *3rd IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA, June 14-16 2006. (Cité page 55.)

E. B. Fowlkes et C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383) :553–569, 1983. (Cité page 100.)

Yoav Freund et Robert E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. Dans *Computational Learning Theory*, volume 904 de *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin / Heidelberg, 1995. (Cité page 109.)

Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, et David Dobkin. Modeling by example. *ACM Transactions on Graphics*, 23 :652–663, August 2004. (Cité page 125.)

Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, et David Dobkin. A search engine for 3D models. *ACM Transactions on Graphics*, 22 :83–105, 2003. (Cité page 81.)

Ran Gal, Ariel Shamir, et Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, 13(2) : 261–271, 2007. (Cité pages 80 et 81.)

Elisa Drelie Gelasca, Touradj Ebrahimi, Mustafa Karaman, et Thomas Sikora. A framework for evaluating video object segmentation algorithms. Dans *Computer Vision and Pattern Recognition Workshop (CVPRW)*, June 2006. (Cité page 103.)

Guido Gerig, Matthieu Jomier, et Miranda Chakos. Valmet : A new validation tool for assessing and improving 3d object segmentation. Dans *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2001*, volume 2208 de *Lecture Notes in Computer Science*, pages 516–523. Springer, 2001. (Cité page 91.)

Daniela Giorgi, Silvia Biasotti, et Laura Paraboschi. SHape Retrieval Contest 2007 : Watertight models track. Dans *SHREC (in conjunction with IEEE Shape Modeling International)*, 2007. (Cité page 94.)

Aleksey Golovinskiy et Thomas Funkhouser. Randomized cuts for 3d mesh analysis. *ACM Transactions on Graphics*, 27(5), December 2008. (Cité pages xiii, 104, 113 et 114.)

Franck Hétroy. Constriction computation using surface curvature. Dans *Eurographics*, pages 1–4, 2005. (Cité pages 77 et 78.)

Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, et Tosiyasu Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. Dans *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 203–212, 2001. (Cité pages 68, 79 et 81.)

Berthold K. P. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12) :1671–1686, December 1984. (Cité page 49.)

Andreas Hubeli et Markus Gross. Multiresolution feature extraction for unstructured meshes. Dans *IEEE Visualization (VIS)*, pages 287–294, 2001. (Cité page 112.)

Evgeny Ivanko et Denis Perevalov. Q–gram statistics descriptor in 3D shape classification. Dans *International Conference on Advances in Pattern Recognition*, pages 360–367, August 2005. (Cité page 49.)

A. K. Jain, M. N. Murty, et P. J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999. (Cité page 44.)

Moonryul Jung et Haengkang Kim. Snaking across 3d meshes. Dans *IEEE Pacific Conference on Computer Graphics and Applications (PG)*, pages 87–93, 2004. (Cité pages 112 et 116.)

Evangelos Kalogerakis, Aaron Hertzmann, et Karan Singh. Learning 3D mesh segmentation and labeling. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4), July 2010. (Cité pages 88, 108, 113 et 119.)

Sang Bing Kang et Katsushi Ikeuchi. Determining 3D object pose using the complex extended gaussian image. Dans *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 580–585, Hawaii, June 3-6 1991. (Cité page 49.)

Sagi Katz, George Leifman, et Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer (Pacific Graphics)*, 21 (8-10) :649–658, 2005. (Cité pages 69, 82, 84 et 104.)

Michael Kazhdan, Thomas Funkhouser, et Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. Dans *Symposium on Geometric Processing*, June 2003. (Cité pages 49 et 50.)

Alireza Khotanzad et Yaw Hua Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5) :489–497, May 1990. (Cité page 39.)

Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69 :86–101, 1967. (Cité page 44.)

Jan J. Koenderink et Andrea J. van Doorn. Surface shape and curvature scales. *Image and Vision Computing*, 10(8) :557–565, October 1992. (Cité pages 38, 75 et 109.)

Yu-Kun Lai, Shi-Min Hu, Ralph R. Martin, et Paul L. Rosin. Fast mesh segmentation using random walks. Dans *ACM Symposium on Solid and Physical Modeling (SPM)*, pages 183–191, 2008. (Cité page 104.)

Guillaume Lavoué, Florent Dupont, et Atilla Baskurt. A new cad mesh segmentation method, based on curvature tensor analysis. *Computer Aided Design*, 37(10) :975–987, 2005. (Cité pages 87, 91, 102 et 115.)

Francis Lazarus et Anne Verroust. Level set diagrams of polyhedral objects. Rapport Technique 3546, Institut National de Recherche en Informatique and en Automatique (INRIA), 1999. (Cité page 68.)

Emilie Lebarbier et Tristan Mary-Huard. Le critère BIC : fondements théoriques et interprétation. Rapport Technique RR5315, INRIA, 2004. (Cité page 45.)

Yong Jae Lee, C. Lawrence Zitnick, et Michael F. Cohen. Shadowdraw : real-time user guidance for freehand drawing. *ACM Transactions on Graphics*, 30 :27 :1–27 :10, August 2011. (Cité page 125.)

Yunjin Lee, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, et Hans-Peter Seidel. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design*, 22(5) :444–465, 2005. (Cité page 112.)

J.P. Lewis, Matt Cordner, et Nickson Fong. Pose space deformations : A unified approach to shape and skeleton-driven deformation. Dans *SIGGRAPH*, pages 165–172, 2000. (Cité page 78.)

Rong Liu et Hao Zhang. Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum*, 26(3) :385–394, 2007. (Cité page 69.)

Shin-Yee Lu et King Sun Fu. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Mans and Cybernetics*, 8(5) :381–389, May 1978. (Cité page 44.)

Saïd Mahmoudi et Mohamed Daoudi. Une nouvelle méthode d'indexation 3D. Dans *13ème Congrès Francophone de Reconnaissance des Formes et Intelligence Articifielle*, volume 1, pages 19–27, Angers, France, January 2002. (Cité page 39.)

Simone Marini, Laura Paraboschi, et Silvia Biasotti. SHape Retrieval Contest 2007 : Partial matching track. Dans *SHREC (in conjunction with IEEE Shape Modeling International)*, pages 13–16, 2007. (Cité page 81.)

David Martin, Charless Fowlkes, Doron Tal, et Jitendra Malik. A database of human segmented natural images and its application to evaluating algorithms and measuring ecological statistics. Dans *International Conference On Computer Vision*, pages 416–423, 2001. (Cité pages 90 et 98.)

Farzin Mokhtarian et Alan K. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1) :34–43, January 1986. (Cité page 39.)

Michela Mortara et Giuseppe Patanè. Affine-invariant skeleton of 3D shapes. Dans *Shape Modeling International*, pages 245–252, 2002. (Cité page 69.)

Chahab Nastar. The image shape spectrum for image retrieval. Rapport Technique 3206, INRIA, 1997. (Cité page 38.)

Xinlai Ni, Michael Garland, et John Hart. Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics, SIGGRAPH*, 23 :613–622, 2004. (Cité page 68.)

Robert Osada, Thomas Funkhouser, Bernard Chazelle, et David Dobkin. Matching 3D models with shape distributions. Dans *International Conference on Shape Modeling and Applications*, May 7-11 2001. (Cité pages 49 et 50.)

Bremer Peer Timo, Herbert Edelsbrunner, Bernd Hamann, et Valerio Pascucci. Topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Volume 10 :385–396, 2004. (Cité page 68.)

Dan Pelleg et Andrew Moore. X-means : Extending k-means with efficient estimation of the number of clusters. Dans *Seventeenth International Conference on Machine Learning*, pages 727–734, 2000. (Cité page 44.)

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American statistical association*, 66(336) :846–850, December 1971. (Cité pages 98 et 99.)

Georges Reeb. Sur les points singuliers d'une forme de pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus de l'Académie des Sciences*, 222 :847–849, 1946. (Cité page 66.)

Julien Ricard, David Coeurjolly, et Atilla Baskurt. Generalizations of angular radial transform for 2D and 3D shape retrieval. *Pattern Recognition Letters*, 26(14) :2174–2186, 2005. (Cité page 49.)

Maximilian Riesenhuber et Thomaso Poggio. Models of object recognition. *Nature neuroscience*, 3 :1199–1204, November 2000. (Cité page 36.)

Bernice E. Rogowitz et Holly E. Rushmeier. Are image quality metrics adequate to evaluate the quality of geometric objects ? Dans *Human Vision and Electronic Imaging*, pages 34–348, 2001. (Cité page 103.)

Pedro V. Sander, John Snyder, Steven J. Gortler, et Hugues Hoppe. Texture mapping progressive meshes. Dans *SIGGRAPH : ACM International Conference on Computer Graphics and Interactive Techniques*, pages 409–416, 2001. (Cité page 91.)

Dietmar Saupe et Dejan V. Vranic. 3D model retrieval with spherical harmonics and moments. Dans *German Association for Pattern Recognition (DAGM)*, pages 392–397, September 2001. (Cité page 49.)

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 :461–464, 1978. (Cité page 45.)

Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6) :1539–1556, 2008. (Cité pages 87 et 89.)

Lior Shapira, Ariel Shamir, et Daniel Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer - International Journal of Computer Graphics*, 24(4) :249–259, 2008. (Cité pages xiii, 102, 104, 109, 113, 114 et 115.)

Alla Sheffer, Emil Praun, et Kenneth Rose. Mesh parameterization methods and their applications. *Foundations and Trends in Computer Graphics and Vision*, 2(2) :105–171, January 2006. (Cité page 87.)

Philip Shilane, Patrick Min, Michael Kazhdan, et Thomas Funkhouser. The Princeton Shape Benchmark. Dans *Shape Modeling International*, June 2004. (Cité pages 36, 48, 49, 50, 52, 53 et 93.)

Jian Sun, Maks Ovsjanikov, et Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5) :1383–1392, 2009. (Cité page 123.)

Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, et Olivier Colot. Deformable shape retrieval using bag-of-feature techniques. Dans *SPIE Electronic Imaging 2011 Symposium (3D Image Processing (3DIP) and Applications II (Conference 7864B))*, San Francisco, California, USA, January 23-27 2011a. (Cité page 123.)

Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, et Olivier Colot. A new 3d-matching method of non-rigid and partially similar models using curve analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4) :852–858, April 2011b. (Cité page 123.)

Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, et Olivier Colot. Non-rigid 3D shape classification using bag-of-feature techniques. Dans *IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, July 11-15 2011c. (Cité page 123.)

Johan W. H. Tangelder et Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. Dans *Shape Modeling International*, June 2004. (Cité page 36.)

Sergey Tarasov et Michael Vyalyi. Construction of contour trees in 3D in O((n) log (n)) steps. Dans *Symposium on Computational Geometry*, pages 68–75, 1998. (Cité pages 67 et 71.)

Adrien Theetten, Tarik Filali Ansary, et Jean-Philippe Vandeborre. 3D-model view characterization using equilibrium planes. Dans *4th International Symposium on 3D Data Processing Visualization Transmission (3DPVT'08)*, Atlanta, Georgia, USA, June 18-20 2008. (Cité page 60.)

Adrien Theetten, Jean-Philippe Vandeborre, et Mohamed Daoudi. Determining characteristic views of a 3D object by visual hulls and Hausdorff distance. Dans *5th IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM 2005)*, Ottawa, Ontario, Canada, June, 13-17 2005. (Cité page 60.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. 3D mesh skeleton extraction using topological and geometrical analyses. Dans *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, Taipei, Taiwan, October 11-13 2006. (Cité page 75.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. Reeb chart unfolding based 3D shape signatures. Dans *Eurographics*, Prague, Czech Republic, September 3-7 2007a. (Cité pages 79 et 81.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. Topology driven 3D mesh hierarchical segmentation. Dans *IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2007)*, Lyon, France, June 13-15 2007b. (Cité pages 83, 87, 102 et 115.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer - International Journal of Computer Graphics*, 24(3) :155–172, March 2008a. (Cité pages 70, 71, 72, 74, 75 et 76.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. Fast and precise kinematic skeleton extraction of 3D dynamic meshes. Dans *19th IEEE International Conference on Pattern Recognition (ICPR-2008)*, Tampa, Florida, USA, December 8-11 2008b. (Cité page 126.)

Julien Tierny, Jean-Philippe Vandeborre, et Mohamed Daoudi. Partial 3D shape retrieval by Reeb pattern unfolding. *Computer Graphics Forum - Eurographics Association*, 28(1) :41–55, March 2009. (Cité pages 79, 81 et 125.)

Ranjith Unnikrishnan, Caroline Pantofaru, et Martial Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE Transaction on pattern analysis and machine intelligence*, 29(6) :929–944, June 2007. (Cité pages 90, 97, 99, 100 et 104.)

Jean-Philippe Vandeborre, Vincent Couillet, et Mohamed Daoudi. A practical approach for 3D model indexing by combining local and global invariants. Dans *1st IEEE International Symposium on 3D Data Processing Visualization Transmission (3DPVT'02)*, Padova, Italy, June, 19-21 2002. (Cité page 38.)

Dejan V. Vranic. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. Dans *IEEE International Conference on Image Processing*, pages 757–760, 2003. (Cité page 49.)

Fu-Che Wu, Wan-Chun Ma, Rung-Huei Liang, Bing-Yu Chen, et Ming Ouhyoung. Domain connected graph : the skeleton of a closed 3D shape for animation. *The Visual Computer*, 22 :117–135, 2006. (Cité page 77.)

Titus Zaharia, Françoise Prêteux, Julien Ricard, Florent Dupont, Atilla Baskurt, Tarik Filali Ansary, Mohamed Daoudi, et Jean-Philippe Vandeborre. Critères pour la comparaison des méthodes d'indexation. Rapport Technique L2.4, Projet RNRT 2002 SEMANTIC-3D, juin 2005. (Cité pages 50 et 51.)

Titus Zaharia et Françoise Prêteux. Three-dimensional shape-based retrieval within the mpeg-7 framework. Dans *SPIE Conference 4304 on Nonlinear Image Processing and Pattern Analysis XII, San Jose, CA, USA*, pages 133–145, January 2001. (Cité page 38.)

Titus Zaharia et Françoise Prêteux. Comparative study for 3D and 2D/3D shape descriptors. Rapport technique, ISO/IEC JTC1/SC29/WG11, MPEG04/10657, München, Germany, March 2004. (Cité page 49.)

Emanoil Zeckerberger, Ayellet Tal, et Shymon Shlafman. Polyhedral surface decomposition with applications. *Computers and Graphics*, 26(5) : 733–743, October 2002. (Cité page 87.)

Hui Zhang, Jason E. Fritts, et Sally A. Goldman. Image segmentation evaluation : A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110 :260–280, 2008. (Cité page 90.)

Annexes

Dans ces annexes, le lecteur trouvera les principales publications dont il est fait référence dans le présent document.

Annexe A, page 144. **A bayesian 3D search engine using adaptive views clustering.** Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. *IEEE Transactions on Multimedia*, 9(1) :78–88, January 2007.

Annexe B, page 156. **Enhancing 3D mesh topological skeletons with discrete contour constrictions.** Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. *The Visual Computer - International Journal of Computer Graphics*, 24(3) :155–172, March 2008.

Annexe C, page 175. **Partial 3D shape retrieval by Reeb pattern unfolding.** Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. *Computer Graphics Forum*, 28(1) :41–55, March 2009.

Annexe D, page 191. **A comparative study of existing metrics for 3D-mesh segmentation evaluation.** Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. *The Visual Computer - International Journal of Computer Graphics*, 26(12) :1451–1466, December 2010.

Annexe E, page 208 **Learning boundary edges for 3d-mesh segmentation.** Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. *Computer Graphics Forum - Eurographics Association*, 30(8) :2170–2182, December 2011.

Annexe F, page 222. **A new 3D-matching method of non-rigid and partially similar models using curve analysis.** Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4) :852–858, April 2011.

Annexe A

A bayesian 3D search engine using adaptive views clustering. Tarik Filali Ansary, Mohamed Daoudi, and Jean-Philippe Vandeborre. *IEEE Transactions on Multimedia*, 9(1) :78–88, January 2007.

Abstract. In this paper, we propose a method for three-dimensional (3-D)-model indexing based on two-dimensional (2-D) views, which we call adaptive views clustering (AVC). The goal of this method is to provide an "optimal" selection of 2-D views from a 3-D model, and a probabilistic Bayesian method for 3-D-model retrieval from these views. The characteristic view selection algorithm is based on an adaptive clustering algorithm and uses statistical model distribution scores to select the optimal number of views. Starting from the fact that all views do not have equal importance, we also introduce a novel Bayesian approach to improve the retrieval. Finally, we present our results and compare our method to some state-of-the-art 3-D retrieval descriptors on the Princeton 3-D Shape Benchmark database and a 3-D-CAD-models database supplied by the car manufacturer Renault.

A Bayesian 3-D Search Engine Using Adaptive Views Clustering

Tarik Filali Ansary, Mohamed Daoudi, *Member, IEEE*, and Jean-Philippe Vandeborre

***Abstract*—In this paper, we propose a method for three-dimensional (3-D)-model indexing based on two-dimensional (2-D) views, which we call adaptive views clustering (AVC). The goal of this method is to provide an "optimal" selection of 2-D views from a 3-D model, and a probabilistic Bayesian method for 3-D-model retrieval from these views. The characteristic view selection algorithm is based on an adaptive clustering algorithm and uses statistical model distribution scores to select the optimal number of views. Starting from the fact that all views do not have equal importance, we also introduce a novel Bayesian approach to improve the retrieval. Finally, we present our results and compare our method to some state-of-the-art 3-D retrieval descriptors on the *Princeton 3-D Shape Benchmark database* and a 3-D-CAD-models database supplied by the car manufacturer *Renault*.**

***Index Terms*—Bayesian approach, clustering, 3–D indexing, 3–D retrieval, views.**

I. Introduction

THE use of three-dimensional (3-D) image and model databases throughout the Internet is growing both in number and size. The development of modeling tools, 3-D scanners, 3-D graphic accelerated hardware, Web3D, and so on, is enabling access to 3-D materials of high quality. In recent years, many systems have been proposed for efficient information retrieval from digital collections of images and videos. However, the solutions proposed so far are not always effective in application contexts where the information is intrinsically 3-D. A similarity metric has to be defined to compute a visual similarity between two 3-D models, given their descriptions.

For example, Kazhdan *et al.* [1] describe a general approach based on spherical harmonics. From the collection of spherical functions calculated on the voxel grid of the 3-D object, they compute a rotation invariant descriptor by decomposing the function into its spherical harmonics and summing the harmonics within each frequency. Then, they compute the L_2-norm for each component. The result is a two-dimensional (2-D) histogram indexed by radius and frequency.

Vandeborre *et al.* [2] propose to use full 3-D information. The 3-D objects are represented as mesh surfaces and 3-D shape descriptors are used. The results obtained show the limitation of the approach when the mesh is not regular. This kind of approach is not robust in terms of shape representation.

Antini *et al.* [3] present an approach based on curvature correlograms. The main advantage of correlograms relates to their ability to encode not only the distribution of features but also their arrangement on the object surface.

Assfalg *et al.* [4] present an approach to global and local content-based retrieval of 3-D models that is based on Spin images. Spin images are used to derive a view-independent description of both database and query objects. A set of Spin images is first created for each object and the parts it is composed of. Then, a descriptor is evaluated for each Spin image in the set. Clustering is performed on the set of image-based descriptors of each object to achieve a compact representation.

Sundar *et al.* [5] encode a 3-D object in the form of a skeletal graph. They use graph matching techniques to match the skeletons and, consequently, to compare the 3-D objects. They also suggest that this skeletal matching approach has the ability to achieve part matching and helps in defining the queries naturally.

In 3-D retrieval using 2-D views, the main idea is that two 3-D models are similar, if they look similar from all viewing angles. Funkhouser *et al.* [6] apply view-based similarity to implement a 2-D sketch query interface. In the preprocessing stage, a descriptor of a 3-D model is obtained by 13 thumbnail images of boundary contours as seen from 13 view directions.

Filali *et al.* [7], [8] propose an adaptive nearest neighbor-like framework to choose the characteristic views of a 3-D model. The framework gives good results but was only applied on a small database.

Using aspect graphs, Cyr and Kimia [9] specify a query by a view of 3-D-objects. A descriptor of a 3-D model consists of a set of views. The number of views is kept small by clustering views and by representing each cluster by one view, which is described by a shock graph. Schiffenbauer [10] presents a complete survey of aspect graphs methods.

Using shock matching, Macrine *et al.* [11] apply indexing using topological signature vectors to implement view-based similarity matching more efficiently.

Chen *et al.* [12] adhere to the intuitive idea that two 3-D models are similar if they also look similar from different angles. They use 100 orthogonal projections of an object and encode them by Zernike moments and Fourier descriptors. They also point out that they obtain better results than other well-known descriptors.

A complete survey on 3-D shape retrieval can be found in Tangelder and Veltkamp [13].

In this paper, we propose a method for 3-D-model indexing based on 2-D views, which we call Adaptive Views Clustering

Manuscript received October 19, 2005; revised March 5, 2006. This work was supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the SEMANTIC-3D National Project (http://liris.cnrs.fr/semantic-3d.net) and in part by the NoE DELOS No. 507618 (http://www.delos.info). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anna Hac.

The authors are with the FOX-MIIRE Research Group (LIFL UMR CNRS/USTL 8022) GET/INT/Télécom, Lille 1, France (e-mail: filali@enic.fr; daoudi@enic.fr; vandeborre@enic.fr).

Color versions of Figs. 1–3, 6– 8, 10, and 14 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2006.886359

1520-9210/$20.00 © 2006 IEEE

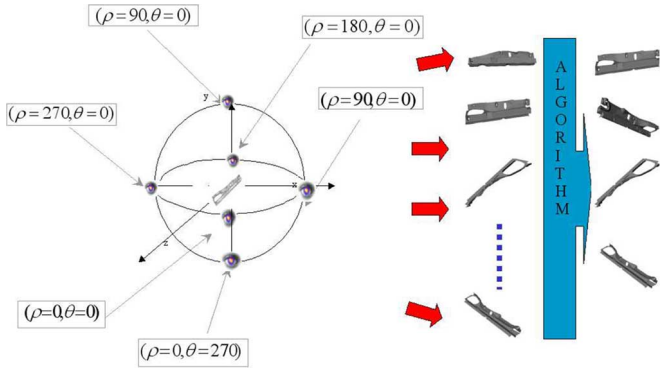


Fig. 1. View selection process.

(AVC). This method aims at providing an optimal selection of 2-D views from a 3-D model, and a probabilistic Bayesian method for 3-D-model indexing from these views. This paper is organized as follows. In Section II, we present the main principles of our method for characteristic view selection. In Section III, our probabilistic 3-D-model indexing is presented. Finally, the results obtained from two databases of 3-D models are discussed demonstrating the performance of our method. We compare our method to some state-of-the-art 3-D retrieval descriptors on the Princeton 3-D Shape Benchmark (PSB) database and the SEMANTIC-3D database.

II. Selection of Characteristic Views

Let $D_b = \{M_1, M_2, \ldots, M_N\}$ be a collection of N 3-D models. We want to represent each 3-D-model M_i by a set of 2-D views that best represent it. Fig. 1 shows an overview of the selection of the characteristic view algorithm. To achieve this goal, we first generate an initial set of views from the 3-D model, then we reduce this set to only those that best characterize this 3-D model. This idea comes from the fact that not all the views of a 3-D model have equal importance: there are views that contain more information than others. For example, one view is sufficient to represent a sphere as it looks the same from all angles. But, more than one view is needed to represent a more complex 3-D model such as an airplane.

In this paragraph, we present our algorithm for characteristic view selection from a 3-D model.

A. Generating the Initial Set of Views

To generate the initial set of views for a model M_i of the collection, we create 2-D views (projections) from multiple viewpoints. These viewpoints are equally spaced on the unit sphere. In our current implementation, we use 320 initial views. In fact, we scale each model such that it can fit into a unit sphere. Then we translate it such that the origin coincides with the 3-D-model barycenter.

To select positions for the views, which must be equally spaced, we use a two-unit icosahedron centred on the origin. We subdivide the icosahedron twice by using the Loop-subdivision schema to obtain a 320 faceted polyhedron. Finally, to generate the initial views, we place the camera on each of the face-centers of the polyhedron looking at the coordinate origin.

The views are silhouettes only, which enhance the efficiency and the robustness of the image metric. Orthogonal projection is applied in order to speed up the retrieval process and reduce the size of the used features. To represent each of these 2-D views, we use 49 coefficients of Zernike moment descriptor [14], [15]. Due to the use of Zernike moments, the approach is robust against translation, rotation, and scaling.

B. Characteristic Views Selection

As every 2-D view is represented by 49 Zernike moment coefficients, choosing a set of characteristic views that best characterize the 3-D models (320 views) is equivalent to choose a subset of points that represent a set of 320 points in a 49-dimension space. Choosing X characteristic views which best represent a set of $N = 320$ views is well known as a *clustering problem*.

One of the widely used algorithm in clustering is K-*means* [16] algorithm. Its attractiveness lies in its simplicity and in its local-minimum convergence properties. However, it has one main shortcoming: the number of clusters K has to be supplied by the user.

As we want from our method to adapt the number of characteristic views to the geometrical complexity of the 3-D model, using K-means is not suitable. To avoid this problem, we use a method derived from K-means. Instead of using a fixed number of clusters, we propose to use a range in which we will choose the "optimal" number of clusters. In our case, the range will be $[1, \ldots, 40]$. In this paper, we assume that the maximum number of characteristic views is 40. This number of views is a good compromise between speed, descriptor size and representation.

We proceed now to demonstrate how to select the characteristic view set and also how to select the best K within the given range. In essence, the algorithm starts with one characteristic view (K equal to 1), we add characteristic views where they are needed, and we do a global K-means on the data starting with characteristic views as cluster centers. We continue alternating between adding characteristic views and doing a global K-means until the upper bound for characteristic view number (40) is reached. During this process, for each K, we save the characteristic view set.

To add new characteristic views, we use the idea presented in the X-means clustering method by Dan Pelleg [17]. First, for every cluster of views represented by a characteristic view, we select two views that have the maximum distance in this cluster. Next, in each cluster of views, we run a local K-means (with $K = 2$) for each pair of selected views. By *local*, we mean that only the views that are in the cluster are used in this local clustering (Fig. 2). Note that Figs. 2 and 3 are just a schematic example, as we represent a view in a 2-D space. In our system, each view is represented by a vector in a 49-dimensional space (corresponding to the 49 Zernike moment features extracted from the view).

At this point, a question arises: "Are the two new characteristic views giving more information on the region than the original characteristic view?" To answer this question, we use Bayesian information criteria (BIC), which scores how likely the representation model (using one or two characteristic views) fits the data. Other criteria, like Akaike information criteria

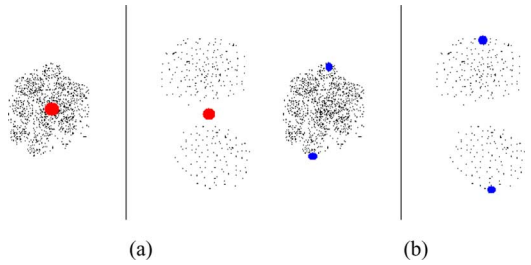


Fig. 2. Local K-means on each part of the views clusters with $K = 2$.

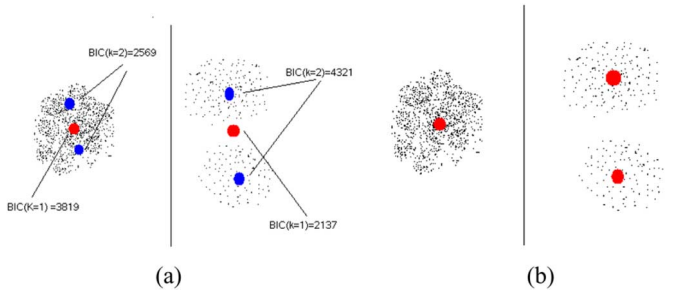


Fig. 3. Selecting the representations (with one or two characteristic views) that have the higher BIC score.

(AIC) [18], could also be used. Estimating the BIC score will be discussed in Section II-C.

According to the outcome of the test, the model with the higher score is selected (Fig. 3). These clusters of the views which are not represented well by the current centroids will receive more attention by increasing the number of centroids in them.

We continue alternating between global K-means and local K-means on clusters belonged to the characteristic views until the upper bound for the characteristic view number is reached. Then, we compare the BIC score of each characteristic view set. Finally, the best characteristic view set will be the one that gets the highest BIC score on all the views. Algorithm 1 gives an overview of the characteristic view selection algorithm.

Algorithm 1 characteristic view selection algorithm.

Number of characteristic views, $K = 1$

while Number of characteristic views $<$ Maximum number characteristic views **do**)

Apply global K-means on all the views (The start centers are the characteristic views).

Save the characteristic view set and its BIC Score.

for all cluster of views **do**

Apply K-means (with $K = 2$) on the cluster.

Choose the representation with the highest BIC score between the original characteristic view or the two new characteristic views.

Update the number of characteristic views.

end for

end while

Select K and the characteristic view set with the higher BIC score.

C. Bayesian Information Criteria

"Representation model selection" refers to the problem of using the data to select one model from a list of candidates $\mathrm{Mod}_1, \ldots, \mathrm{Mod}_k$ that best represent the data. In our case, a representation model will be a set of characteristic views.

Model selection is a well known mathematical problem. This problem can be solved using many criteria like AIC [18], BIC [19], Bayes Factors [20] and so on. The Bayesian solution to these problems is to compute the posterior probability $P(\mathrm{Mod}_j|D)$ for each model. Under weak conditions, BIC and Bayes methods are asymptotically equivalent. On the other hand, it appears to be some debate on the relative merits of AIC versus BIC, but it is beyond the scope of this paper [21], [22]. In practice, it has been observed that BIC selects models with a dimension smaller than AIC, because BIC "penalizes" more than AIC when dimension is higher than seven [23]. In this paper, we use the BIC for model selection.

To calculate the BIC score for a *representation model* Mod_j having the cluster of views V, we use the formula introduced by Schwarz [19]

$$\mathrm{BIC}(\mathrm{Mod}_j) = \hat{l}_j(V) - \frac{P_j}{2}\log N \tag{1}$$

where P_j is the number of parameters in Mod_j also known as the Schwarz criterion [19], $\hat{l}_j(V)$ is the log-likehood of the data according to the jth model taken at the maximum likelihood point, and N ($N = |V|$) is the number of views in the cluster.

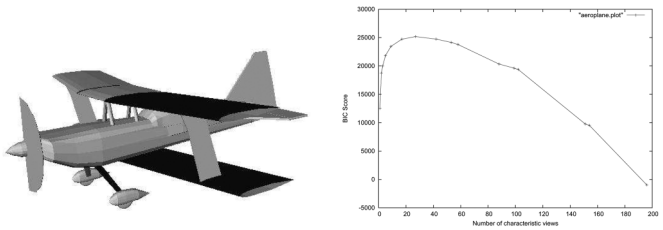
In our case the models are all spherical Gaussians which is the type assumed by K-means. The maximum likehood estimate (MLE) for variance is

$$\hat{\theta}^2 = \frac{1}{N-K}\sum_i (\mathrm{Dist}(V_i, Vc_i)^2) \tag{2}$$

where $\mathrm{Dist}(V_i, Vc_i)$ the Euclidean distance between the Zernike moments of the respective views V_i and Vc_i. Here, Vc_i is the characteristic view associated with the view V_i. The log-likehood of the data is

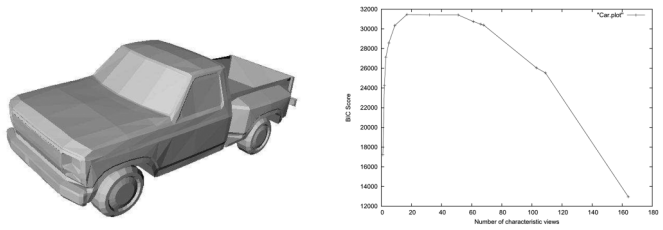
$$\hat{l}_j(V) = \sum_i \left(\frac{1}{\sqrt{2\pi\hat{\theta}^{49}}} - \frac{1}{2\hat{\theta}^2} \| \mathrm{Dist}(V_i, Vc_i)\|^2 + \log\frac{N_{(i)}}{N} \right). \tag{3}$$

Figs. 4 and 5 show the evolution of the BIC score with the number of views. These curves show that an "optimal" number of views exists where the BIC score is maximized. For the airplane model in Fig. 4(a), the number of optimal views is 29. For the car model in Fig. 5(a), only 17 views are needed as this 3-D model is less complex than the first one. This results from



(a) (b)

Fig. 4. (a) Airplane model from the PSB database and (b) its corresponding BIC score curve.



(a) (b)

Fig. 5. (a) Car model from the PSB and (b) its corresponding BIC Score curve.

the fact that as the geometrical complexity of the 3-D model increases, the number of distinct 2-D-views also increases. This leads to a higher number of views to best represent it.

III. Probabilistic Approach for 3-D Indexing

As mentioned before, not all views of a 3-D model have the same importance. There are views which represent the 3-D model better than others. On the other hand, simple objects (e.g., cube, sphere) can be at the root of more complex objects. In this section, we present a probabilistic approach that takes into account that views do not have the same importance, and that simple objects have higher probability to appear than more complex one.

Each model of the collection $D_b = \{M_1, M_2, \ldots, M_N\}$ is represented by a set of characteristic views $V_c = \{V_1, V_2, \ldots, V_{\hat{v}}\}$, with $\hat{v}$ being the number of characteristic views. To each characteristic view corresponds a set of represented views called V_r.

Considering a 3-D-request-model Q, we want to find the model $M_i \in D_b$ which is the closest to the request model Q. This model is the one that has the highest probability $P(M_i|Q)$.

Knowing that each model is represented by its characteristic views, $P(M_i|Q)$ can be written

$$P(M_i|Q) = \sum_{k=1}^{K} P\left(M_i|V_Q^k\right) P\left(V_Q^k|Q\right) \tag{4}$$

where K is the number of characteristic views of the model Q.

Let H^k be the set of all the possible hypotheses of correspondence between the request view V_Q^k and a model M_i, $H^k = h_1^k \vee h_2^k \vee \cdots \vee h_N^k$. A hypothesis h_p^k is a Boolean variable which denotes that the view p of the model corresponds or not to the view request V_Q^k. The sign $\vee$ represents *the logical or operator*. Let us note that if an hypothesis h_p^k is true, then all the other hypotheses are false. $P\left(M_i|V_Q^k\right)$ can be expressed by $P\left(M_i|H^k\right)$.

We have

$$P\left(M_i|H^k\right) = \sum_{j=1}^{\hat{v}} P\left(M_i, V_{M_i}^j|H^k\right) \tag{5}$$

$$P(M_i|H^k) = \sum_{j=1}^{\hat{v}} \sum_{p=1}^{\hat{v}} P\left(M_i, V_{M_i}^j|h_p^k\right). \tag{6}$$

Since we cannot observe in the same scene two characteristic views of the same 3-D model, we have

$$P\left(M_i, V_{M_i}^j|h_p^k\right) = 0 \text{ for every } j \neq p$$

then

$$P\left(M_i|H^k\right) = \sum_{j=1}^{\hat{v}} P\left(M_i, V_{M_i}^j|h_j^k\right). \tag{7}$$

The sum $\sum_{j=1}^{\hat{v}} P\left(M_i, V_{M_i}^j|h_j^k\right)$ can be reduced to the only true hypothesis $P\left(M_i, V_{M_i}^j|h_j^k\right)$. In fact, a characteristic view from the request model Q can match only one characteristic view from the model M_i. We choose the characteristic view with the maximum probability

$$P\left(M_i|H^k\right) = \text{Max}_j P\left(M_i, V_{M_i}^j|h_j^k\right). \tag{8}$$

By (4) and (8), we have

$$P(M_i|Q) = \sum_{k=1}^{K} \text{Max}_j(P(M_i, V_{M_i}^j|h_j^k)) P\left(V_Q^k|Q\right). \tag{9}$$

The closest model is the one that contains the view having the highest probability.

Using the Bayes theorem, we obtain

$$P\left(M_i, V_{M_i}^j|h_j^k\right) = \frac{P\left(h_j^k, V_{M_i}^j|M_i\right) P(M_i)}{P\left(h_j^k\right)}. \tag{10}$$

On the other hand, we have

$$P\left(h_j^k, V_{M_i}^j|M_i\right) = P\left(h_j^k|V_{M_i}^j, M_i\right) P\left(V_{M_i}^j|M_i\right) \tag{11}$$

and

$$P\left(h_j^k\right) = \sum_{i=1}^{N} \sum_{j=1}^{\hat{v}} P\left(h_j^k|V_{M_i}^j, M_i\right) P\left(V_{M_i}^j|M_i\right) P(M_i). \tag{12}$$

By using this remark, we obtain

$$P\left(M_i, V_{M_i}^j|h_j^k\right) = \frac{P\left(h_j^k|V_{M_i}^j, M_i\right) P\left(V_{M_i}^j|M_i\right) P(M_i)}{\sum_{i=1}^{N} \sum_{j=1}^{\hat{v}} P\left(h_j^k|V_{M_i}^j, M_i\right) P\left(V_{M_i}^j|M_i\right) P(M_i)}. \tag{13}$$

Finally, we have (14), as shown at the bottom of the page.

As mentioned in the begining of this section, the main idea of our probabilistic approach is that not all 3-D models in the collection have the same probability to occur. Simple objects (e.g., cube, sphere) could be at the root of more complex objects. Our algorithm assumes that the simpler is the 3-D model, the smaller is the number of the characteristic views. To model the fact that the larger the relative number of views of a model M_i, the smaller the probability of the model, we estimate $P(M_i)$, the probability to observe a 3-D model M_i by

$$P(M_i) = \frac{e^{(-\alpha N(V_{M_i})/N(V))}}{\sum_{i=1}^{N} e^{(-\alpha N(V_{M_i})/N(V))}} \tag{15}$$

where $N(V_{M_i})$ is the number of characteristic views of the model M_i, $N(V)$ is the total number of characteristic views for the set of the models of the collection D_b. α is a coefficient that reduces the effect of small values of the exponentiel in $P(M_i)$.

On the other hand, there are views that contain more information than other ones. We assume that the greater the number of represented views $N(Vr^j_{M_i})$ for a characteristic view $V^j_{M_i}$, the more this characteristic view is important and the more information it contains about the 3-D model. So, we modeled $P(V^j_{M_i}|M_i)$ the probability to observe the characteristic view j of the model M_i by

$$P\left(V^j_{M_i}|M_i\right) = \frac{1-\beta e^{\left(-\beta N\left(Vr^j_{M_i}\right)/N(Vr_{M_i})\right)}}{\sum_{j=1}^{\hat{v}}\left(1-\beta e^{\left(-\beta N(Vr^j_{M_i})/N(Vr_{M_i})\right)}\right)} \tag{16}$$

where $N\left(Vr^j_{M_i}\right)$ is the number of views represented by the characteristic view j of the model M and $N(Vr_{M_i})$ is the total number of views represented by the model M_i. The β coefficient is introduced to reduce the effect of small values of the view probability. We used different values for α and β in our experiments to study the evolution of our algorithm.

The value $P\left(h^k_j|V^j_{M_i}, M_i\right)$ is the probability that, knowing that we observe the characteristic view j of the model M_i, this view corresponds to the view k of the request model Q

$$P\left(h^k_j|V^j_{M_i}, M_i\right) = \frac{e^{-D\left(V^k_Q, V^j_{M_i}\right)}}{\sum_{j=1}^{\hat{v}} e^{-D\left(V^k_Q, V^j_{M_i}\right)}} \tag{17}$$

where $D\left(V^k_Q, V^j_{M_i}\right)$ is the Euclidean distance between the 2-D Zernike descriptors of the view k of the request model Q and $V^j_{M_i}$ is the characteristic view j of the 3-D model M_i.

To summarize, in this section we presented our Bayesian retrieval framework which takes into account the number of characteristic views of the model and the importance (amount of information) of its views.

IV. Experimental Setup

In this section, we present the experimental results, the databases and the evaluation criteria. The algorithms we described in the previous sections have been implemented using C++ and the TGS OpenInventor libraries. The system consists of an off-line characteristic view extraction algorithm and an online retrieval process.

In our AVC method, every model was normalized for size by isotropically rescalling it so that the average Euclidean distance from points on its surface to the center of mass is 0.5. Then, all models were normalized for translation by moving their center of mass to the origin.

In the off-line process, the characteristic view selection takes about 18 s per model on a PC with a Pentium IV 2.4-GHz CPU. In the online process, the comparison takes less than 1 s for 1814 3-D models. To evaluate our method, we used two different 3-D-model databases. The PSB is a standard shape benchmark widely used in shape retrieval community. We also present our results on a more specific database, 5000 3-D CAD models provided by the car manufacturer Renault within the framework of SEMANTIC project [24].

During our experiments, to show the contribution of the probabilistic approach to retrieval performance, we compare the probabilistic correspondence between two 3-D models to the use of a simple distance between the 3-D models. We define a dissimilarity between a 3-D-model M_i and a query Q as

$$\mathrm{Dist}(M_i, Q) = \sum_{j=1}^{\hat{v}} Min_k\, D(V^j_{M_i}, V^k_Q),$$

where $D\left(V^j_{M_i}, V^k_Q,\right)$ is the Euclidean distance between the 2-D Zernike descriptors of the view k of the request model Q and $V^j_{M_i}$ the characteristic view j of the 3-D model M_i.

A. Data Set

Sections IV-A1 and A2 present the 3-D-models databases we used in our experiments.

1) Princeton Shape Benchmark: PSB appeared in 2004 and is one of the most exhaustive benchmarks for 3-D shape retrieval. It contains a database of 1814 classified 3-D models collected from 293 different Web domains. There are many classifications given to the objects in the database. During our experiments we used the finest granularity classification, composed of

$$P(M_i|Q) = \sum_{k=1}^{K} \mathrm{Max}_j \left(\frac{P\left(h^k_j|V^j_{M_i}, M_i\right) P\left(V^j_{M_i}|M_i\right) P(M_i)}{\sum_{i=1}^{N}\sum_{j=1}^{\hat{v}} P\left(h^k_j|V^j_{M_i}, M_i\right) P\left(V^j_{M_i}|M_i\right) P(M_i)} \right) P\left(V^k_Q|Q\right). \tag{14}$$

161 classes. Most classes contain objects with a particular function (e.g., cars). Yet, there are also cases where objects with the same function are partitioned in different classes based on their shapes (e.g., round tables versus rectangular tables) [25]. We compare our method to some state-of-the-art shape-matching algorithms.

- **D2 Shape Distribution:** a histogram of distances between pairs of points on the surface [26].
- **Extended Gaussian Image (EGI):** a spherical function giving the distribution of surface normals [27].
- **Complex Extended Gaussian Image (CEGI):** a complex-valued spherical function giving the distribution of normals and associated normal distances of points on the surface [28].
- **Q-Gram Statistics Descriptor (2-GR):** a vector in 256-dimensional space, whose i-th co-ordinate is the number of 2-grams Q_i in the voxel representation of 3-D shape [29].
- **Shape Histogram (SHELLS):** a histogram of distances from the center of mass to points on the surface [30].
- **Shape Histogram (SECTORS):** a spherical function giving the distribution of model area as a function of spherical angles [30].
- **Shape Histogram (SECSHELL):** a collection of spherical functions that give the distribution of model area as a function of radius and spherical angles [30].
- **Voxel:** a binary rasterization of the model boundary into a voxel grid which is represented by the 32 spherical descriptors giving the intersection of the voxel grid with the concentric spherical shells.
- **Spherical Extent Function (EXT):** a spherical function giving the maximal distance from center of mass as a function of the angle [31].
- **Radialized Spherical Extent Function (REXT):** a collection of spherical functions giving the maximal distance from the center of mass as a function of spherical angle and radius [32].
- **Gaussian Euclidean Distance Transform (GEDT):** a 3-D function whose value at each point is given by the composition of a Gaussian with the Euclidean Distance Transform of the surface [1].
- **Spherical Harmonic Descriptor (SHD):** a rotation invariant representation of the GEDT obtained by computing the restriction of the function to cocentric spheres and storing the norm of each (harmonic) frequency [1].
- **Light Field Descriptor (LFD):** a representation of a model as a collection of images rendered from uniformly sampled positions on a view sphere. The distance between two descriptors is defined as the minimum $L1$ distance, taken over all rotations and all pairings of vertices on two dodecahedra [12].

Every model was normalized for size by isotropically rescaling it so that the average distance from points on its surface to the center of mass is 0.5. Then, for all descriptors except D2 and EGI, the model was normalized for translation by moving its center of mass to the origin. Next, for all descriptors except D2, SHELLS, SHD and LFD, the model was normalized for rotation by aligning its principal axes to the x-, y-, and z-axes. The ambiguity between positive and negative axes was resolved by choosing the direction of the axes so that the area of model on the positive side of the x-, y-, and z-axes was greater than the area on the negative side. Every spherical descriptor (EGI, CEGI, Sectors, etc.) was computed on a 64×64 spherical grid and represented by its harmonic coefficients up to order 16th order. Histograms of distances (D2 and Shells) were stored with 64 bins representing distances in the range $[0, 2]$. All descriptors, except LFD, were scaled to have L_2-norm equal to 1. The LFD comprises 100 images encoded with 35 8-bit coefficients to describe Zernike moments and ten 8-bit coefficients to represent Fourier descriptors. For more details about the PSB database, refer to [25] (Fig. 6).

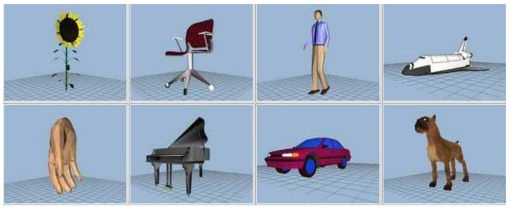


Fig. 6. 3-D models from PSB database [25].

2) SEMANTIC-3D Database: The SEMANTIC project [24], supported by the French Research Ministry and the RNRT aims to explore the techniques and tools preliminary to the realization of new services for the exploitation of 3-D contents through the Web. New techniques of compression, indexing and watermarking of 3-D data are developed and implemented in a industrial prototype application. These exchanges are ensured with a format of data standardized with functionalities of visualization and 3-D animation. The exchanges are achieved on existing networks (internal network of the authors, wireless networks for the technicians, etc.).

In SEMANTIC project, our research group focuses on the indexing and retrieval of 3-D-data.

Within this framework, a database of 3-D CAD models was provided by the car manufacturer Renault which database contains 5000 of quite irregular polygonal meshes representing CAD parts. To measure the performance, we classified 758 models of this collection into 75 classes based on the judgment of two adult persons (Fig. 7), the 4242 other models were put in a class called "others". The smallest class contains four 3-D models. The largest class contains sixty-four 3-D models. Fig. 7 shows some 3-D models from the SEMANTIC Database.

B. Evaluation Criterion

To objectively evaluate our method we use a statistical tool provided by PSB to compare 3-D retrieval methods. Given a classification and a distance matrix computed with any shape matching algorithm, a suite of PSB tools produces statistics and visualizations that facilitate the evaluation of the match results [25].

The statistics that can be calculated using the PSB tools are:

- **Nearest neighbor:** the percentage of the closest matches that belong to the same class as the query. This statistic provides an indication of how well a nearest neighbor classifier would perform. Obviously, an ideal score is 100%, Higher scores represent better results.

Fig. 7. 3-D models from SEMANTIC database.

- **First-tier and Second-tier:** the percentage of models in the query's class that appear within the top K matches, where K depends on the size of the query's class. Specifically, for a class with $|C|$ members, $K = |C| - 1$ for the first tier, and $K = 2 \times (|C| - 1)$ for the second tier. The first tier statistic indicates the recall for the smallest K that could possibly include 100% of the models in the query class, while the second tier is a little less stringent (i.e., K is twice as big). These statistics are similar to the Bull Eye Percentage Score $K = 2 \times |C|$, which has been adopted by the MPEG-7 visual SDs. In all cases, an ideal matching result gives a score of 100%, again higher values indicate better matches.
- **E-Measure:** a composite measure of the precision and recall for a fixed number of retrieved results. The intuition is that a user of a search engine is more interested in the first page of query results than in later pages. So, this measure considers only the first 32 retrieved models for every query and calculates the precision and recall over those results. The E-Measure is defined as:

$$E = \frac{2}{\frac{1}{P} + \frac{1}{R}}.$$

  The E-measure is equivalent to subtracting van Rijsbergen's definition of the E-measure from 1. The maximum score is 1.0, and higher values indicate better results
- **Discounted Cumulative Gain (DCG):** a statistic that weights correct results near the front of the list more than correct results later in the ranked list under the assumption that a user is less likely to consider elements near the end of the list. Specifically, the ranked list R is converted to a list G, where element G_i has a value 1 if element R_i is in the correct class and 0 otherwise. DCG is then defined as follows:

$$\mathrm{DCG}_1 = G_1; \mathrm{DCG}_i = \mathrm{DCG}_{i-1} + \frac{G_i}{lg_2(i)}, \text{ if } i > 1.$$

  The result is then divided by the maximum possible DCG (i.e., that would be achieved if the first C elements were in the correct class, where C is the size of the class) to give the final score

$$\mathrm{DCG} = \frac{\mathrm{DCG}_k}{1 + \sum_{j=2}^{|C|} \frac{1}{lg_2(j)}}$$

  where k is the number of models in the database. The entire query result list is incorporated in an intuitive manner by the DCG [33].
- **Recall versus Precision Curves:** Recall versus Precision Curves are well known in the literature of content-based search and retrieval. The recall and precision are defined as follow:

$$\mathrm{Recall} = \frac{N}{Q}, \quad \mathrm{Precision} = \frac{N}{A}$$

  where N is the number of relevant models retrieved in the top A retrievals, Q is the number of relevant models in the collection, which is the number of models to which the query belongs to.

V. Experiments and Results

A. Princeton Shape Benchmark

In our experiments, we computed the distances between all pairs of models in the PSB database and analyze them with the PSB evaluation tools to quantify the matching performance with respect to the base classification.

Fig. 8 shows a request using our 3-D retrieval system. On the left side, the request 3-D model is presented. The right side shows the 3-D models which have the highest probabilities of matching the 3-D request model.

As mentioned before, we use several performance measures to objectively evaluate our method: FT, ST, NN, DCG, and N-DCG match percentages, as well as the recall-precision plot [34].

Table I shows micro-average storage requirements (for our method, we used 23 views which is the average number of characteristic views for all the database models) and retrieval statistics for each algorithm. Storage size is given in bytes. We found that micro and macro-average results gave consistent results, and we decided to present micro-averaged statistics.

Fig. 9 shows the Recall versus Precision plots for our AVC method and some other shape descriptors. For visual clarity, we presented the curves of the most relevant algorithm only.

We find that the shape descriptors based on 2-D views (LFD and AVC) provide the best retrieval precision in this experiment. We might expect shape descriptors that capture 3-D geometric relationships would be more discriminating than the ones based solely on 2-D projections; however, the opposite is true.

Our method provides more accurate results with the use of Bayesian probabilistic indexing. The experiment shows that AVC gives better performance than 3-D harmonics, Radialized Spherical Extent Function, and Gaussian Euclidean Distance Transform on the PSB database. Light Field Descriptor gives better results than our method but uses 100 views, does not adapt the number of views to the geometrical complexity, and uses two descriptors for each view (Zernike moments and

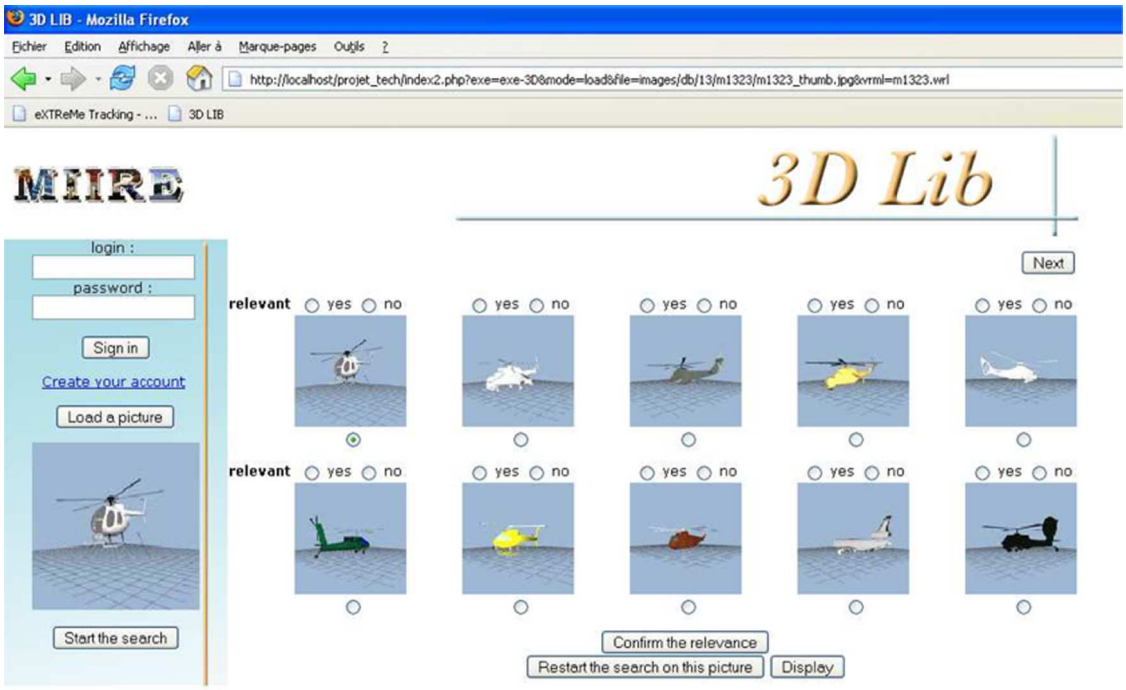


Fig. 8. Screenshot of the 3-D-model retrieval system.

TABLE I
RETRIEVAL PERFORMANCES FOR PSB

Methods	Discrimination						
	Storage size	NN	FT	ST	E-Measure	DCG	N-DCG
LFD	4,700	65.7%	38.0%	48.7%	28.0%	64.3%	21.3%
AVC(probability)	**1,113**	**60.6%**	**33.2%**	**44.3%**	**25.5%**	**60.2%**	**13.48%**
REXT	17,416	60.2%	32.7%	43.2%	25.4%	60.1%	13.3%
GEDT	32,776	60.3%	31.3%	40.7%	23.7%	58.4%	10.2%
AVC(simple distance)	**1,113**	**58.2%**	**31.1%**	**42.7%**	**25.1%**	**59.9%**	**11,8%**
SHD	2,184	55.6%	30.9%	41.1%	24.1%	58.4%	10.2%
2-GR	512	55.5%	28.7%	39.1%	23.0%	56.3%	—%
EXT	552	54.9%	28.6%	37.9%	21.9%	56.2%	6.0%
SECSHEL	32,776	54.6%	26.7%	35.0%	20.9%	54.5%	2.8%
VOXEL	32,776	54.0%	26.7%	35.3%	20.7%	54.3%	2.4%
SECTORS	552	50.4%	24.9%	33.4%	19.8%	52.9%	-0.3%
CEGI	2,056	42.0%	21.1%	28.7%	17.0%	47.9%	-9.6%
EGI	1,032	37.7%	19.7%	27.7%	16.5%	47.2%	-10.9%
D2	136	31.1%	15.8%	23.5%	13.9%	43.4%	-18.2%
SHELLS	136	22.7%	11.1%	17.3%	10.2%	38.6%	-27.3%

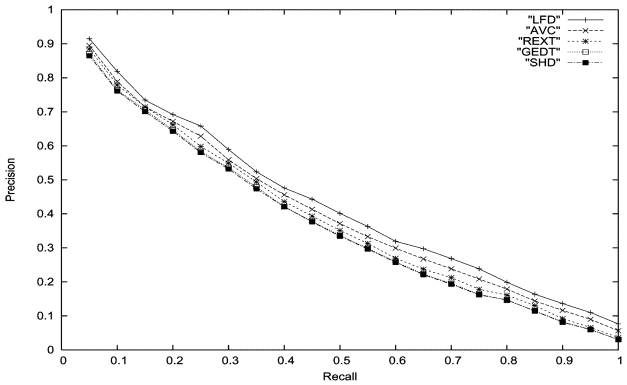


Fig. 9. Recall precision on PSB database.

Fourier descriptor), which makes it a slower and more memory consuming compared to the method we have presented.

Overall, we conclude that AVC gives a good compromise between quality (relevance) and cost (memory and online comparison time) for the 3-D models of the PSB.

B. SEMANTIC-3D Database

The experiments are made on a database that contains 5000 3-D models. To objectively evaluate the performance of our method on this database, a classification was made (ground truth). 758 models are classified on 75 classes. A ”special” class called *others* contains all the 3-D models that were not classified.

Fig. 10(a) shows a 3-D model from SEMANTIC database that is used as a query to our system. Fig. 10(b) shows the ranked results to the query in Fig. 10(a).

In Section III, we presented our probabilistic approach that uses two different parameters α and β. We conducted experiments on the SEMANTIC database using different values of α

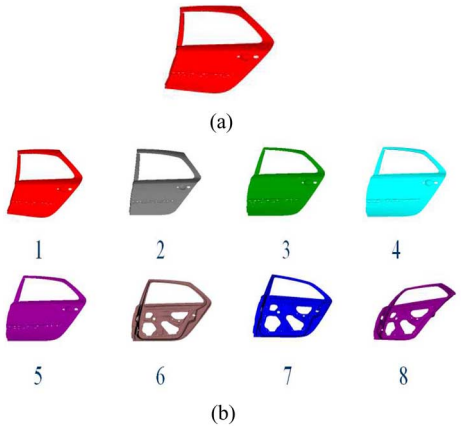


Fig. 10. (a) Query 3-D model and the (b) results on SEMANTIC database.

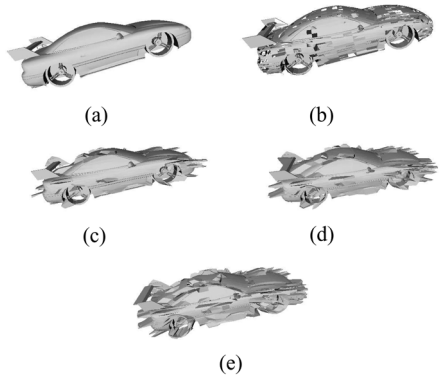


Fig. 12. Robustness evaluation of noise and decimation from a 3-D model. (a) Orginal 3-D model. (b) Model with decimation. (c) Model with 5% noise. (d) Model with 10% noise. (e) Model with 15% noise.

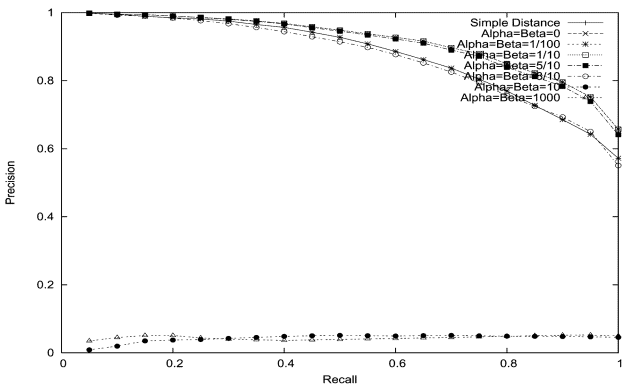


Fig. 11. Recall versus Precision for SEMANTIC database.

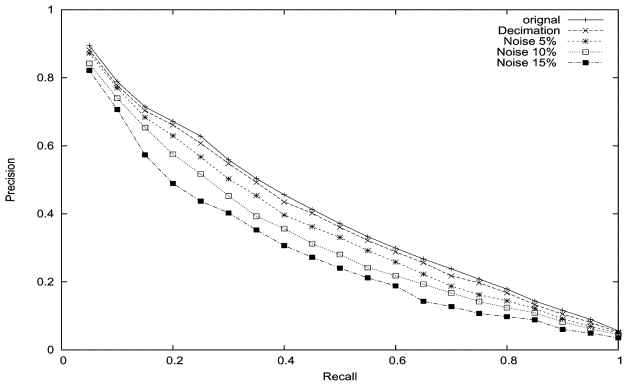


Fig. 13. Recall versus Precision on PSB database with deformations effects.

and β. Different Recall versus Precision plots have been computed (Fig. 11) for different α and β coefficients values.

Fig. 11 shows the Recall versus Precision plots for our AVC method on the SEMANTIC database. Some remarks can be achieved on these plots. First of all, when $\alpha = \beta = 0$, the Recall versus Precision plot is the same as the one for our *AVC simple distance*. This behavior could be explained easily by replacing in the (15) and (16) the values of α and β by 0. When α and β coefficients become greater than 1 the Recall versus Precision plots looks like a random retrieval. The variation of α and β coefficients between 0 and 1 gives better Recall versus Precision plots and scores. Experimentally, as Recall versus Precision plots in Fig. 11 show, the best results are obtained for $\alpha = \beta = 1/100$. That is why this value has been choosen for all experiments.

We can explain the good results of our method by the fact that the intra-class variance are very small. As the database contains real professional 3-D-CAD models, the 3-D models from the same class represent real mechanical automobile parts. The 3-D models from the same class represent different versions of the same models with small changes.

On the other hand, 3-D-CAD models in the database contain holes so that they can be attached to other mechanical parts. The positions and the dimension of the holes can differentiate between two different models from the same class. As we represent each view of the 3-D model by Zernike moments, the holes and the global shape are well taken into account. We can also notice the result enhancement when we use the probabilistic approach for retrieval.

C. Robustness Evaluation of the Method

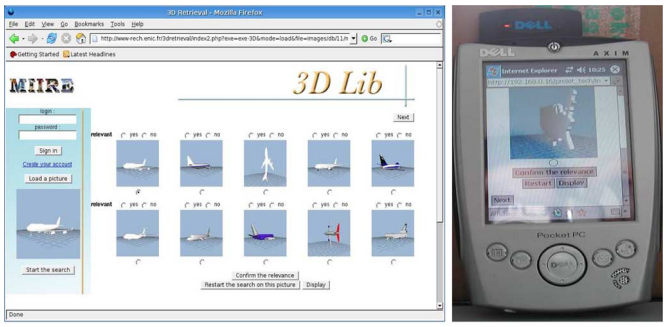
In order to assess the robustness of our method, we apply the following transformations for all classified 3-D models. Each transformed 3-D model is then used to query the test database.

The average recall and precision of all the 1814 classified models from the PSB database are used for the evaluation (Fig. 13). The robustness is evaluated by the following transformation.

1) Noise: to 15% of the 3-D model vertices we applied three random translations in the directions of x-, y-, and z-axis ($\pm 5\%$, 10% and 15% the length of the model's bounding box). Fig. 12(c)–(e) show typical examples of the noise effect.
2) Decimation: for each 3-D model, randomly select 20% polygons to be deleted. Fig. 12(b) shows a typical example of the decimation effect.

In Fig. 13, the first plot represents the original Recall versus Precision without any decimation nor noise addition. Decimation [Fig. 12(b)] gives very close results to the ones with the original models.

The Fig. 12 shows the effects of noise addition for different values 5%, 10%, and 15% on a 3-D-model; 5% noise addition could easily be obtained in real situations such as the use of a 3-D scanner to get a 3-D-model from a real object.



(a) (b)

Fig. 14. (a) PC and (b) PDA browser interface of our online search engine.

Precision versus Recall plots were computed for each value of noise and for decimation (Fig. 13). From these results, it can be observed that the performance of our approach is robust to 5% noise addition and to decimation. Even if 10% and 15% are high noise addition, Precision versus Recall plots (Fig. 13) shows that our approach remains quite robust. The 10% and 15% noise curve shows the "limits" of our method in terms of noise robustness.

D. On-Line Search Engine

The SEMANTIC project focuses on the development of tools and methods required to implement new operational services for retrieving 3-D content through the Web and communicating objects. Information and communication system must be available for remote access and assistance, interconnecting originators (mechanical part designers), nomadic users (automotive industry technicians) and a central 3-D data server.

To test our algorithms and to assess the results presented in the previous sections, we developed an online 3-D search engine. Our search engine can be reached from any device having a compatible web browser (PC, PDA, SmartPhone, etc.) [35].

Fig. 14(a) shows the interface of our system using a PC web browser. Fig. 14(b) shows the interface of our system on PDA (Pocket PC under Windows Pocket 2003).

Depending on the web access device he/she is using, the user faces two different kinds of web interfaces: a rich web interface for full-featured web browsers [Fig. 14(a)], and a simpler interface for PDA web browsers [Fig. 14(b)]. In both cases, the results returned by the 3-D search engine are the same. The only difference lies in the design of the presentation of results.

Finally, the users should notice that, due to some copyright protection, the 3-D-CAD models from the SEMANTIC project are not available for public online use. The 3-D-database available for tests of our 3-D search engine is the PSB database [25].

VI. Discussion and Conclusion

In this paper, we propose a 3-D-model retrieval system based on characteristic view similarity which we called AVC. Starting from the fact that the more the 3-D model is geometrically complex, the more its 2-D views are different, we propose a characteristic view selection algorithm that relates the number of views to its geometrical complexity. Starting from 320 initial views, our algorithm selects the "optimal" characteristic view set that best represents the 3-D model. The number of characteristic views varies from 1 to 40. We also propose a new probabilistic retrieval approach that takes into account that not all the views of 3-D models have the same importance, and also the fact that geometrically simple models have more probability to be relevant than more complex ones. Based on some standard measures, we compared our method to thirteen state-of-the-art methods on PSB database. Our method gives the second-best results. The AVC method we proposed gives a good quality/cost compromise compared to other well-known methods. The results of our method on a large 3-D-CAD-model database (5000 models) supplied by *Renault*, show that our method can also be suitable for 3-D-CAD-model retrieval. Our method is robust against noise and model degeneracy. It is usable in the case of topologically ill-defined 3-D model. A practical 3-D-model retrieval system based on our approach is available on the web for online tests [35].

Acknowledgment

The authors would like to thank Professor D. Pelleg from Carnegie Mellon University for the help he provided on X-means.

References

[1] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proc. Symp. Geometry Processing*, Aachen, Germany, 2003.

[2] J. P. Vandeborre, V. Couillet, and M. Daoudi, "A practical approach for 3D model indexing by combining local and global invariants," in *Proc. IEEE Int. Symp. 3D Data Processing, Visualization and Transmission*, Padova, Italy, Jun. 2002, pp. 644–647.

[3] G. Antini, S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval of 3D objects using curvature correlograms," in *Proc. IEEE Int. Conf. Multimedia & Expo*, Tokyo, Japan, Jul. 2005.

[4] J. Assfalg, A. Del Bimbo, and P. Pala, "Spin images for retrieval of 3D objects by local and global similarity," in *Proc. IEEE Int. Conf. Pattern Recognition*, Aug. 2004, vol. 3, pp. 906–909.

[5] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proc. IEEE Shape Modeling Int.*, 2003.

[6] T. Funkhouser, P. Min, M. Kazhdan, A. Haderman, D. Dobkin, and D. Jacobs, "A search engine for 3D models," *ACM Trans. Graph.*, vol. 22, no. 1, pp. 83–105, 2003.

[7] T. F. Ansary, J. P. Vandeborre, and M. Daoudi, "Bayesian approach for 3D models retrieval based on characteristic views," in *Proc. IEEE Int. Conf. Pattern Recognition*, Cambridge, U.K., Aug. 2004.

[8] T. F. Ansary, J. P. Vandeborre, S. Mahmoudi, and M. Daoudi, "A Bayesian framework for 3D models retrieval based on characteristic views," in *Proc. IEEE Int. Symp. 3D Data Processing, Visualization and Transmission*, Thessaloniki, Greece, Sep. 2004.

[9] C. M. Cyr and B. Kimia, "3D object recognition using shape similarity-based aspect graph," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 254–261.

[10] R. D. Schiffenbauer, A Survey of Aspect Graphs CIS, 2001, Tech. Rep. TR-CIS-2001-01.

[11] D. Macrini, A. Shokoufandeh, S. Dickenson, K. Siddiqi, and S. Zucker, "View based 3-D object recognition using shock graphs," in *Proc. IEEE Int. Conf. Pattern Recognition*, 2002, vol. 3, pp. 24–28.

[12] D. Y. Chen, X. P. Tian, Y. T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," in *Eurographics*, 2003, vol. 22, pp. 223–232.

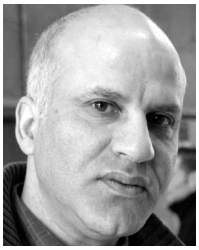
[13] J. W. H. Tangelder and R. C. Veltkamp, "A survey of content-based 3D shape retrieval methods," in *Proc. IEEE Shape Modeling Int.*, 2004, pp. 145–156.

[14] A. Khotanzad and Y. H. Hong, "Invariant image recognition by zernike moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 5, pp. 489–497, May 1990.

[15] W. Y. Kim and Y. S. Kim, "A region-based shape descriptor using zernike moments," *Signal Process.: Image Commun.*, vol. 16, pp. 95–100, 2000.
[16] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[17] D. Pelleg and A. Moore, "X-means: Extending K-means with efficient estimation of the number of clusters," in *Proc. Int. Conf. Machine Learning*, 2000, pp. 727–734.
[18] H. Akaike, "Information theory and extention of the maximum likelihood principle," in *IEEE Int. Symp. Information Theory*, 1973, pp. 267–281.
[19] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.
[20] J. O. Berger and L. Pericchi, "The intrinsic Bayes factor for model selection and prediction," *J. Amer. Statist. Assoc.*, pp. 109–122, 1994.
[21] K. P. Burnham and D. Anderson, *Model Selection and Multi-model Inference*. New York: Springer-Verlag, 2002.
[22] H. Bozdogan, "Model selection and Akaike's information criteria (AIC): The general theory and its analytical extentions," *Psychometrika* , vol. 52, pp. 354–370, 1987.
[23] E. Lebarbier and T. Mary-Huard, Le Critère BIC: Fondements théoriques et interprétation INRIA, 2004, Tech. Rep. RR5315.
[24] Semantic Project [Online]. Available: http://www.semantic-3d.net
[25] Princeton Shape Benchmark 2004 [Online]. Available: http://shape.cs.princeton.edu/benchmark
[26] R. Osada, T. Funkhouser, B. Chazells, and D. Dobkin, "Matching 3D models with shape distributions," in *Proc. IEEE Shape Modeling Int.*, May 2001.
[27] B. Horn, "Extended Gaussian images," *Proc. IEEE 72*, vol. 12, pp. 1671–1686, Dec. 1984.
[28] S. Kang and K. Ikeuchi, "Determining 3D object pose using the complex extended gaussian image," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 1991, pp. 580–585.
[29] E. Ivanko and D. Pervalov, "Q-gram statistics descriptor in 3D shape classification," in *Proc. Int. Conf. Advances in Pattern Recognition*, Aug. 2005, pp. 360–367.
[30] M. Ankerst, G. Kastenmuller, H. P. Kriegel, and T. Seidl, "Nearest neighbor classification in 3D protein databases," in *Int. Soc. Behav. Med.*, 1999.
[31] D. Saupe and D. V. Vranic, "3D model retrieval with spherical harmonics and moments," in *German Assoc. Pattern Recognit. (DAGM)*, Sep. 2001, pp. 392–397.
[32] D. V. Vranic, "An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres," in *Proc. IEEE Int. Conf. Image Processing*, 2003, pp. 757–760.
[33] G. Leifman, S. Katz, A. Tal, and R. mei, "Signatures of 3d models for retrieval," in *4th Israel-Korea Bi-Nat. Conf. Geometric Modeling and Computer Graphics*, Aug. 2003, pp. 159–163.
[34] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *IEEE Shape Modeling Int.*, 2004.
[35] Fox-Miire—3D Search Engine [Online]. Available: http://www-rech.enic.fr/3dretrieval

Tarik Filali Ansary received the B.E. degree from Ecole Marocaine des Sciences de l'Ingénieur, Morocco, in 2002, and the M.S. degree from the University of Sciences and Technologies of Lille (USTL), Lille, France, in 2003, both in computer science. He received the Ph.D. degree in 2006 from the Institut National des Télécommunications, France, and at USTL with the FOX-MIIRE research group at the LIFL (UMR USTL/CNRS 8022), a Research Laboratory in the computer science field of the USTL.

His principal research interests include 3-D content-based retrieval, multimedia indexing, and retrieval from content.



Mohamed Daoudi (M'97) received the Ph.D. degree in computer science from the University of Sciences and Technologies of Lille (USTL), Lille, France, in 1993 and the accreditation to supervise research (HDR) in computer science from the University of Littoral, Littoral, France, in 2000.

Since 1994, he has been an Associate Professor in the Computer Science Department, INT/Télécom Lille 1, where he teaches courses on data structures, distributed algorithms, and computer vision. From 2004 to 2005, he was a Professor at the Ecole Polytechnique de Tours. From 2000 to 2004, he was the Head of the MIIRE group of LIFL (UMR USTL/CNRS 8022), a Research Laboratory in the computer science field of the USTL. His research interests include digital pattern recognition, image processing, invariant representation of images and shapes, neural network, and three-dimensional content-based retrieval. He has contributed to more than 60 articles in journals and conference proceedings.

Dr. Daoudi has served as a Program Committee member for the International Conference on Pattern Recognition (ICPR) in 2004 and the International Conference on Multimedia and Expo (ICME) in 2004 and 2005. He is a frequent reviewer for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and for *Pattern Recognition Letters*.

Jean-Philippe Vandeborre received the M.S. degree in 1997 and the Ph.D. degree in computer science in 2002, both from the University of Sciences and Technologies of Lille (USTL), Lille, France.

Currently, he is an Assistant Professor of the Groupe des Ecoles des Télécommunications (GET) at INT/Télécom Lille 1 (a graduate engineering school in information technology and communications), France, in the Computer Science and Network Department where he teaches TCP/IP networks, Unix systems, and computer graphics. He is also a member of the LIFL (UMR USTL/CNRS 8022), a Research Laboratory in the Computer Science field of the USTL. His current research interests are mainly focused on 3-D model analysis, and include multimedia indexing and retrieval from content and their applications.

Annexe B

Enhancing 3D mesh topological skeletons with discrete contour constrictions. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. *The Visual Computer - International Journal of Computer Graphics*, 24(3) :155–172, March 2008.

Abstract. This paper describes a unified and fully automatic algorithm for Reeb graph construction and simplification as well as constriction approximation on triangulated surfaces. The key idea of the algorithm is that discrete contours – curves carried by the edges of the mesh and approximating the continuous contours of a mapping function – encode both topological and geometrical shape characteristics. Therefore, a new concise shape representation, enhanced topological skeletons, is proposed, encoding the contours' topological and geometrical evolution. First, mesh feature points are computed. Then they are used as geodesic origins for the computation of an invariant mapping function that reveals the shape most significant features. Next, for each vertex in the mesh, its discrete contour is computed. As the set of discrete contours recovers the whole surface, each of them can be analyzed, both to detect topological changes and constrictions. Constriction approximations enable Reeb graphs refinement into more visually meaningful skeletons, which we refer to as enhanced topological skeletons. Extensive experiments showed that, without any preprocessing stage, proposed algorithms are fast in practice, affine-invariant and robust to a variety of surface degradations (surface noise, mesh sampling and model pose variations). These properties make enhanced topological skeletons interesting shape abstractions for many computer graphics applications.

Visual Comput (2008) 24: 155–172
DOI 10.1007/s00371-007-0181-0

ORIGINAL ARTICLE

Julien Tierny
Jean-Philippe Vandeborre
Mohamed Daoudi

Enhancing 3D mesh topological skeletons with discrete contour constrictions

Published online: 9 November 2007
© Springer-Verlag 2007

J. Tierny (✉)
LIFL (UMR USTL/CNRS 8022),
University of Lille, France
julien.tierny@lifl.fr

J.P. Vandeborre · M. Daoudi
GET/TELECOM Lille 1
LIFL (UMR USTL/CNRS 8022),
University of Lille, France
{jean-philippe.vandeborre,
mohamed.daoudi}@lifl.fr

Abstract This paper describes a unified and fully automatic algorithm for Reeb graph construction and simplification as well as constriction approximation on triangulated surfaces.
The key idea of the algorithm is that *discrete contours* – curves carried by the edges of the mesh and approximating the continuous contours of a mapping function – encode both topological and geometrical shape characteristics. Therefore, a new concise shape representation, *enhanced topological skeletons*, is proposed, encoding the contours' topological and geometrical evolution.
First, mesh feature points are computed. Then they are used as geodesic origins for the computation of an invariant mapping function that reveals the shape most significant features. Next, for each vertex in the mesh, its *discrete contour* is computed. As the set of discrete contours recovers the whole surface, each of them can be analyzed, both to detect topological changes and constrictions. Constriction approximations enable Reeb graphs refinement into more visually meaningful skeletons, which we refer to as *enhanced topological skeletons*. Extensive experiments showed that, without any preprocessing stage, proposed algorithms are fast in practice, affine-invariant and robust to a variety of surface degradations (surface noise, mesh sampling and model pose variations). These properties make *enhanced topological skeletons* interesting shape abstractions for many computer graphics applications.

Keywords Shape abstraction · Topological skeletons · Feature points · Constrictions · Topology driven segmentation

1 Introduction

Polygonal mesh is a widely used representation of 3D shapes, mainly for exchange and display purposes. However, many applications in computer graphics need higher level shape descriptions as input. Topological skeletons have shown to be interesting shape descriptions [4]. They are beneficial for diverse fields like shape metamorphosis [25], deformation [8], retrieval [16], texture mapping [35], and so on.

Many topological approaches study the properties of real valued functions computed over triangulated surfaces. Most of the time, those functions are provided by the application context, such as scientific data analysis [7]. When dealing with topological skeletons, it is necessary to define an invariant and visually interesting mapping function, which remains an open issue [4].

Moreover, traditional topological graph construction algorithms assume that all the information brought by the mapping function is pertinent, while in practice, this can lead to large graphs [9, 24], encoding noisy details.

Finally, topological approaches cannot discriminate visually interesting subparts of identified connected compo-

nents, like the phalanxes of a finger. This is detrimental to certain applications, such as mesh deformation.

In this paper, an original and unified framework is proposed to address the above issues. Given a closed connected triangulated surface T, feature points are firstly extracted (Fig. 1a) in order to compute an invariant mapping function, noted f_m (Fig. 1b), which reveals the shape most significant parts. Secondly, for each vertex in the mesh, we compute its discrete contour, a connected curve traversing it and approximating f_m continuous contours. We show that a topological analysis of those discrete contours enables a pertinent Reeb graph construction and simplification (Fig. 1c), without any input parameter. Finally, we show that a geometrical analysis of discrete contours can approximate constrictions on prominent components (Fig. 1d), enabling the refinement of Reeb graphs into enhanced topological skeletons (Fig. 1e).

This paper, which extends the authors' previous work [32], is structured as follows. First, we introduce topological skeleton related work. Next, we define our mapping function f_m. Then, we present our algorithm for discrete contour computation, which is used both for the Reeb graph construction and simplification as well as the constriction approximation.

In particular, special emphasis is given to the discrete contour constriction detection algorithm, which provides more robust and more accurate results than the previous work [32]. Moreover, an extensive discussion on experimental results and framework evaluation is proposed. Finally, to demonstrate the usability of the presented method, two applications are discussed: topology driven mesh segmentation (Fig. 1f) and skeleton driven mesh deformation (Fig. 1g).

Fig. 1a–g. Main steps of the framework. First, feature points are extracted (**a**). Then a mapping function (**b**) is contoured both to construct a topological skeleton (**c**) and to enhance it (**e**) with constrictions (**d**). Topology driven segmentation (**f**) and skeleton driven deformation (**g**) are two examples of application of the framework

2 Related work

Several approaches have been explored for the decomposition of polygonal meshes into meaningful subcomponents, to extract skeletal representations of shapes. They can be roughly classified into four categories: semantic-oriented segmentations, medial axis transforms, generalized potential fields and topological skeletons. Each one has its own advantages and drawbacks. For example, medial axis transform based methods [6, 26] are commonly associated with the drawback of high sensitiveness to small surface perturbations, which is detrimental to many applications. On the contrary, generalized potential field based methods [33] provide more robust results at the cost of high computational times. In comparison to mesh segmentation based methods [2, 17, 18] and traditional skeleton extraction [33], topological approaches, based on Morse and Reeb graph theories [22, 28], present the advantage of preserving the topological properties of the shape [4] (number of loops, number and relations between components, etc.). However, with regard to shape skeletons, we identify three main drawbacks in topological approaches, successively addressed in this paper.

Firstly, it is difficult to define an invariant and visually interesting mapping function. Secondly, constructing and transforming a topological graph into a manageable skeleton is not a trivial problem. Finally, topological approaches decompose a surface into connected subcomponents only. This means that visually interesting subparts of identified connected components will not be discriminated: for example, a finger of a hand model will not be decomposed into phalanxes.

2.1 Mapping functions

Differential topology based approaches study the properties of real valued functions, which we refer to as *mapping functions*, defined on input surfaces, either to construct Reeb graphs [10, 30], contour trees [9], level set diagrams [20] or Morse complexes [7, 24]. Those functions are often brought by the application context: terrain modeling [30], MRI analysis [9], molecular analysis [7], etc.

When dealing with topological skeletons, it is necessary to define a scalar function which satisfies invariance and stability constraints, and which also provides a topological description that highlights visually significant surface subcomponents.

Lazarus and Verroust [20] introduced such a function, defined by the *geodesic distance* (the length of the shortest

path between vertices) from a source vertex to any other vertex in the mesh. It leads to visually interesting results for natural objects because it is invariant to geometrical transformations and it is robust against variations in model pose [17]. Due to a lack of stability, within the framework of shape retrieval, Hilaga et al. [16] proposed to integrate this function all over the mesh. Unfortunately, from our experience, that function generates an important amount of critical points, configurations where the gradient of the function vanishes, which makes the construction of visually meaningful graphs more complex.

In our method, to reveal the most significant features, we focus on feature points. Feature points are mesh vertices located on extremities of prominent components [17]. Mortara and Patanè [23] proposed to select as feature points the vertices where Gaussian curvature exceeds a given threshold, but this cannot resolve extraction on constant curvature areas. Katz et al. [17] developed an algorithm based on multi-dimensional scaling in quadratic execution complexity. In this paper, we propose a robust and straightforward algorithm for feature point extraction (Fig. 1a). Moreover, we use them as geodesic origins for the definition of our mapping function (Fig. 1b). Such a function reveals well the most visually significant parts of the mesh, generating manageable critical point sets.

2.2 Graph construction and simplification

A Reeb graph [28] is a topological structure that encodes the connectivity relations of the critical points of a scalar function defined on an input surface. More formally, Reeb graphs are defined as follows:

Definition 1 (Reeb graph). *Let $f : M \to \mathbb{R}$ be a Morse function defined on a compact manifold M. The Reeb graph of f is the quotient space of f in $M \times \mathbb{R}$ by the equivalence relation $(p_1, f(p_1)) \sim (p_2, f(p_2))$, which holds iff:*

$$\begin{cases} f(p_1) = f(p_2) \\ p_1 \text{ and } p_2 \text{ belong to the same connected} \\ \text{component of } f^{-1}(f(p_1)). \end{cases}$$

Figure 2 gives an example of a Reeb graph computed on a bi-torus with regard to the height function and illustrates well the fact that Reeb graphs can be used as skeletons.

Constructing a Reeb graph from a scalar function f computed on a triangulated surface first requires to identify the set of vertices corresponding to critical points. With this aim, several formulations have been proposed [11, 31] to identify local maxima, minima and saddles, observing for each vertex the evolution of f at its direct neighbors. Several algorithms have been developed to construct Reeb graphs from the connectivity

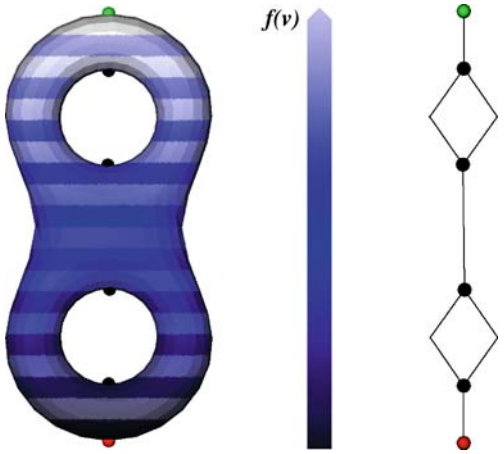


Fig. 2. Evolution of the level lines of the height function on a bi-torus, its critical points and its Reeb graph

relations of these critical points [9, 10], most of them in $O(n \times \log(n))$ steps, with n the number of vertices in the mesh. However, they assume that all the information brought by the scalar function f is relevant [9, 24]. Consequently, they assume that all the identified critical points are meaningful, while in practice, this hypothesis can lead to unmanageably large Reeb graphs. To overcome this issue, Ni et al. [24] developed a user-controlled simplification algorithm. Bremer et al. [7] proposed an interesting critical point cancellation technique based on a *persistence* threshold. Attene et al. [1] proposed a seducing approach, unifying the graph construction and simplification, but it is conditioned by a *slicing* parameter.

In this paper, we propose a discrete formulation of contours, connected subsets of level lines, which enables, without any input parameter, the construction of visually meaningful Reeb graphs (Fig. 1c).

2.3 Constriction computation

Psychological research [5] claims that the human visual system tends to segment complex objects along the narrowest and the most concave regions. Based on this hypothesis, in order to improve the topological skeleton description quality, we focus on constrictions on surfaces.

Hétroy and Attali [15] define constrictions as simple closed curves, whose length is locally minimal. Recently, Hétroy [14] showed that constriction detection could be achieved by analyzing surface curvature.

In this paper, we propose to analyze the geometrical characteristics of discrete contours, and particularly their curvature, to approximate constrictions (Fig. 1d), in order to decompose previously identified components into more visually interesting parts (Fig. 1e).

3 Framework overview

Given a closed connected triangulated surface T, we propose in this paper a unified method to decompose T into visually meaningful subparts, considering the topological and geometrical characteristics of *discrete contours*.

The algorithm proceeds in three stages. First, mesh feature points are extracted (Fig. 1a) in order to compute an invariant and visually interesting mapping function (Fig. 1b), denoted as f_m in the rest of the paper. Next, for each vertex in the mesh, we compute its discrete contour, a curve traversing it and approximating f_m continuous contour. Finally, as the set of discrete contours recovers the entire mesh, it is possible to analyze each contour's characteristics, either to detect topological changes (Fig. 1c) or to detect curvature transitions (Fig. 1d).

Our scientific contribution resides in three points: *(1)* We propose a robust and straightforward algorithm for feature point extraction. *(2)* We show that a discrete contour formulation enables, without re-meshing and without any input parameter, a pertinent Reeb graph construction, providing visually meaningful graphs, affine-invariant and robust to surface degradations. *(3)* We show that the geometrical information brought by discrete contours enables the approximation of constrictions on prominent components and consequently Reeb graph refinement.

4 Feature point extraction

To compute visually meaningful topological skeletons, we first have to define a mapping function that will highlight the most significant parts of the mesh. To achieve this, we first focus on feature points, vertices located on the extremities of prominent components, because they provide a good overview of the shape structure [17, 23].

4.1 Algorithm overview

To extract feature points, we propose a quite straightforward algorithm, using differential topology tools. It is based on the key idea that feature points can be extracted analyzing some mapping function local extrema. In Fig. 3a, the height function has been computed on a hand model. As the fingers are directed upwards, local maxima (in green) are identified at the extremities of related components and can be consequently referred to as feature points. However, it is straightforward that the height function is not a relevant choice in the general case because it is dependent on the object orientation.

The function that maps a vertex to its geodesic distance to a source vertex seems much more appropriate because geodesic distances are affine-invariant and robust to variations in model pose. In Fig. 3b, c the geodesic distance function has been computed on the hand model, using the extremity of the middle finger (in red) as source vertex. The set of local extrema (minima and maxima, in red and green) actually contains the set of feature points (extremities of the fingers and the wrist).

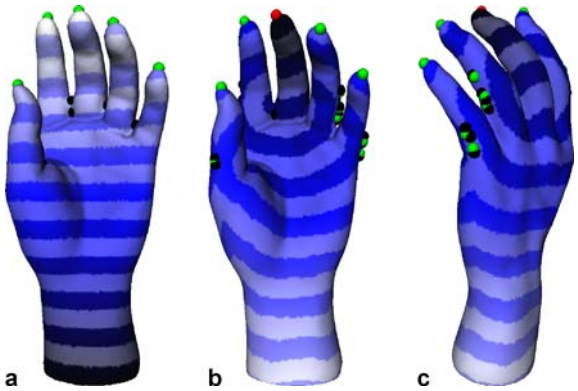


Fig. 3a–c. Euclidean (**a**) and geodesic (**b**,**c**) based mapping function critical points

However, Fig. 3c shows some local maxima in configurations which do not correspond to feature points (green points on the side of the little and the ring fingers). To discriminate local extrema that correspond to feature points from those which do not, we propose to realize a crossed analysis, using two geodesic based mapping functions – whose origins are the mesh's most distant vertices – and to intersect the sets of their local extrema.

4.2 Algorithm formulation

From an algorithmic point of view, geodesic distances can be approximated by the Moore–Dijkstra algorithm (distance minimizing in weighted graphs). In the rest of this paper, we will refer to $\delta(v_i, v_j)$ as the normalized approximation of the geodesic distance from vertex v_i to v_j, normalized with regard to mesh global extrema.

Let v_{s_1} and v_{s_2} be the most geodesic distant vertices of a closed connected triangulated surface T, computed with the tree diameter algorithm [20]. In Fig. 4, v_{s_1} is located at

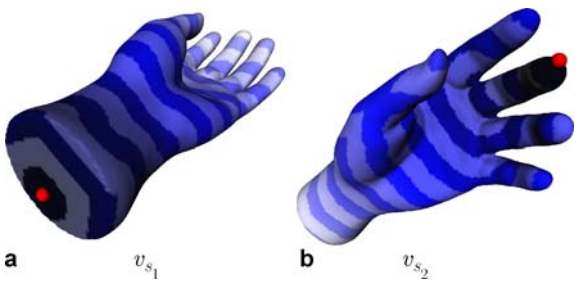


Fig. 4a,b. Most geodesic distant vertices of the hand model

the extremity of the wrist (Fig. 4a) while v_{s_2} is located at the extremity of the middle finger (Fig. 4b).

Let f_{g_1} and f_{g_2} be two scalar functions defined on each vertex v of T, as follows:

$$f_{g_1}(v) = \delta(v, v_{s_1}) \tag{1}$$

$$f_{g_2}(v) = \delta(v, v_{s_2}). \tag{2}$$

Based on the critical point classification proposed in [10], a *local minimum* is defined as a vertex such that all its direct neighbors have an upper function value. Reciprocally, we define a *local maximum* as a vertex such that all its direct neighbors have a lower function value.

Let E_1 be the set of local extrema (minima and maxima) of f_{g_1} (in yellow in Fig. 5a) and E_2 be the set of local extrema of f_{g_2} (in cyan in Fig. 5b). Extremities of prominent components are configurations where f_{g_1} and f_{g_2} tend to an extremum (Fig. 5a, b). Consequently, the set of feature points is both included in E_1 and E_2. Therefore, we define the set of feature points F of T (Fig. 5c) as follows:

$$F = E_1 \cap E_2 \,. \tag{3}$$

In practice, f_{g_1} and f_{g_2} local extrema which correspond to feature points do not appear exactly on the same vertices but in the same *geodesic neighborhood*. Therefore, the intersection constraint is relaxed as follows, with $\epsilon \in [0, 1]$ the radius of the geodesic neighborhood (geodesic distances are normalized):

$$v \in F \Longleftrightarrow \begin{cases} \exists v_{e_1} \in E_1 \quad / \quad \delta(v, v_{e_1}) < \epsilon \\ \exists v_{e_2} \in E_2 \quad / \quad \delta(v, v_{e_2}) < \epsilon \\ \delta(v, v_{f_i}) > \epsilon \quad \forall v_{f_i} \in F \\ \qquad\qquad \epsilon \in [0, 1]. \end{cases} \tag{4}$$

From our experience, using only two geodesic mapping functions (f_{g_1} and f_{g_2}) and setting $\epsilon = 0.05$ gives accurate results. The choice of such settings as well as the efficiency and properties of the presented algorithm will be discussed in Sect. 10.

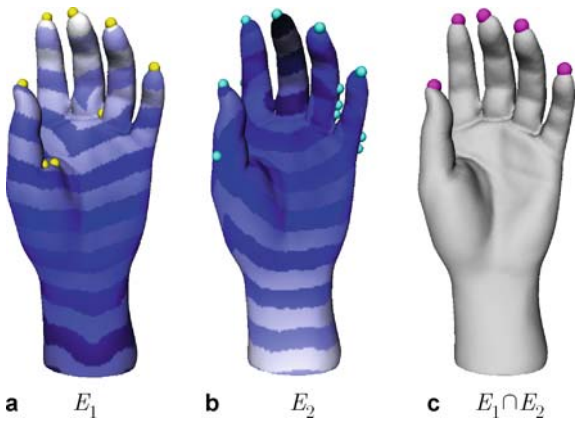


Fig. 5a–c. Feature point extraction overview

5 Mapping function definition

The definition of the mapping function depends on what is expected to be revealed. For example, for terrain modeling [30], the *height function* will present critical points over hills and valleys, providing consequently an appropriate topological description. When dealing with topological skeletons, it is necessary to define an invariant and visually interesting mapping function, which highlights the global structure of the object. Moreover, the mapping function should not generate an unmanageable set of critical points, in order to make the graph simplification easier. From our experience, this is not the case of the function presented in [16].

Firstly, to guarantee invariance to geometrical transformations and robustness against variations in model pose, geodesic distances are used. Secondly, to define a visually interesting mapping function, feature points are taken as origins for geodesic distance evaluations, because they provide a good overview of the object global structure [23].

Therefore, we propose the following mapping function, denoted as f_m in the rest of the paper, which computes in each vertex v of T the geodesic distance to the closest feature point:

$$f_m(v) = \frac{f_c(v) - \min_{v \in T} f_c(v)}{\max_{v \in T} f_c(v) - \min_{v \in T} f_c(v)}. \tag{5}$$

The mapping function f_m is a normalized version of the function f_c, defined as follows ($f_c(v) \geq 0, \forall v \in T$):

$$f_c(v) = 1 - \delta'(v, v_c), \tag{6}$$

with v_c the closest feature point from v:

$$v_c \in F \quad / \quad \delta'(v, v_c) = \min_{v_{f_i} \in F} \delta(v, v_{f_i}). \tag{7}$$

Notice that f_m is invariant to uniform scaling (thanks to the normalization), rotation and translation (thanks to the use of geodesic distances). Figure 6 presents some computations of f_m over arbitrary shapes, the number of extracted feature points ($|F|$) and the number of critical points ($|C|$, identified according to the classification proposed in [10]). The term f_m has been defined so that it tends to maxima (in green) at feature points and tends to minima (in red) at the center of the object.

As shown in Fig. 6, f_m generates an important number of critical points. Consequently, standard Reeb graph construction algorithms would create large graphs, counting as many nodes as critical points, which is a major issue for the extraction of meaningful topological skeletons.

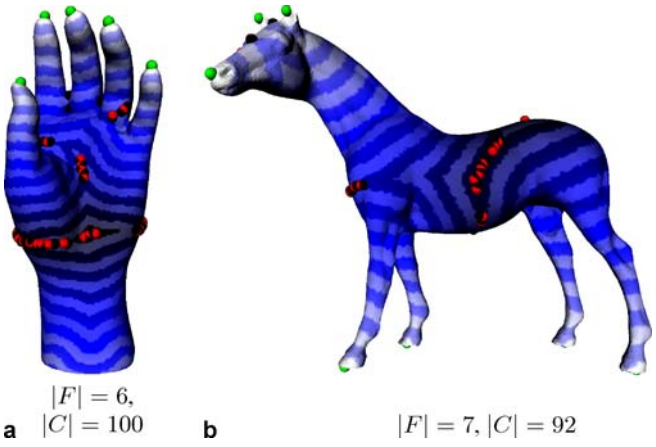


Fig. 6a,b. The f_m mapping function computed on two standard models. The important number of identified critical points prevents standard algorithms from producing concise shape representations

In the next section, we present a formulation of discrete contours, which enables a unified graph construction and simplification process.

6 Discrete contour computation

Defining contours of a real function f computed on a triangulated surface T is not a simple problem. In the continuous case, a level line $f^{-1}(f(p))$ is the set of points p_i such that $f(p_i) = f(p)$. Moreover, two points p_1 and p_2 belong to the same *contour* if they belong to the same connected component of $f^{-1}(f(p_1))$.

In the discrete case, for a given vertex $v \in T$, depending on T sampling, the set of vertices v_i such that $f(v_i) = f(v)$ is often reduced to the vertex v itself. With regard to Definition 1, a correct Reeb graph could not be constructed from this formulation of discrete contours, because the conditions of the equivalence relation would rarely be satisfied.

To preserve contour topological properties in the discrete case, we define the *discrete level line* $\Gamma(v)$ associated with the vertex v as a curve computed along the edges of T which approximates by upper value the continuous level line $f^{-1}(f(v))$. More formally, it can be defined as follows:

Definition 2 (Discrete level line). *Let T^+ be the subset of a closed connected triangulated surface T such that $\forall v^+ \in T^+$, $f(v^+) \geq f(v)$. The discrete level line associated with the vertex v and denoted as $\Gamma(v)$ is the set of edges (and related vertices) belonging to T^+, such that each edge of $\Gamma(v)$ is adjacent to only one face of T^+.*

The discrete level line $\Gamma(v)$ is the boundary between the upper valued triangulation T^+ and the rest of T. Figure 7 shows discrete level lines traversing an arbitrary triangulation, with regard to the height function. Moreover, each connected subset of a discrete level line is referred to as a discrete contour.

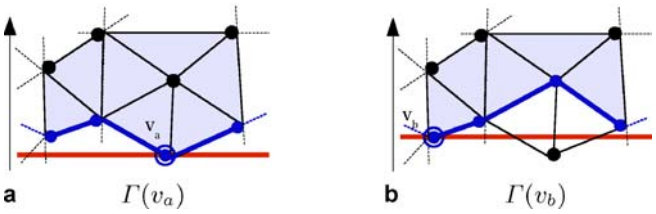


Fig. 7a,b. Example of continuous (*red*) and discrete (*blue*) level lines (height function). The upper valued triangulation T^+ is colored in *light blue*

In particular, we define the discrete contour $\gamma(v)$ associated with the vertex v as the connected subset of $\Gamma(v)$ containing v. Notice that as T becomes more and more dense, the more discrete contours will tend to continuous contours.

Discrete contours can be computed for the whole mesh using a step by step gradient ascent process, described in Algorithm 1. This algorithm describes a propagation from f global minimum to f local maxima. It handles two heaps, respectively, the set of visited vertices Vt and the set of candidate vertices for visit Cd. At each step, Cd surrounds Vt by an upper value and corresponds to the discrete level line $\Gamma(v)$, with $v = \text{argmin}_{v' \in Cd} f(v')$.

Algorithm 1. Discrete contour computation

$Vt = \emptyset$
$Cd \leftarrow \{\text{argmin}_{v' \in T} f(v')\}$
while $Cd \neq \emptyset$ **do**
  $v \leftarrow \text{argmin}_{v' \in Cd} f(v')$
  $\Gamma(v) \leftarrow Cd$
  $\gamma(v) \leftarrow$ connected subset of Cd, containing v
  Remove v from Cd
  $Cd \leftarrow Cd \cup \{v$ neighbors, which are not in $Vt\}$
  Add v to Vt
end while

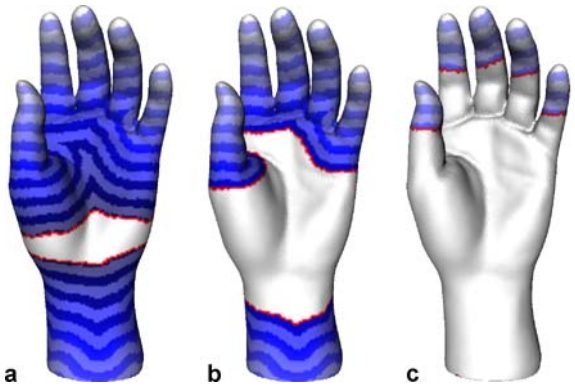


Fig. 8a–c. Examples of discrete level lines (in *red*) on a 26 000 vertex mesh (f_m function). $\Gamma(v_{2000})$ (**a**), $\Gamma(v_{10\,000})$ (**b**) and $\Gamma(v_{20\,000})$ (**c**) are respectively composed of 2, 3 and 6 *discrete contours*

In Fig. 8, examples of discrete level lines $\Gamma(v_i)$ are shown, at different iterations i of the algorithm. The Vt vertex set is displayed in white and $\Gamma(v)$ is displayed in red. Visiting in a recursive fashion each vertex of $\Gamma(v)$ enables the identification of each of its connected subsets, and particularly $\gamma(v)$. In the next sections, discrete contours will be analyzed, both to detect topological changes and constrictions.

7 Topological analysis of discrete contours

Standard Reeb graph construction algorithms need simplification in order to remove noisy details. In this section, we propose a unified algorithm for graph construction and simplification, based on the topological analysis of discrete contours. Following the Definition 1 of a Reeb graph in the continuous case, we can state an analog equivalence relation in the discrete case between two vertices $v_1, v_2 \in T$, based on our notion of discrete contour:

$$(v_1, f(v)) \sim (v_2, f(v)) \Longleftrightarrow \begin{cases} v_1, v_2 \in \Gamma(v) \\ v_1, v_2 \in \gamma(v). \end{cases} \quad (8)$$

The terms v_1 and v_2 belong to the same connected component if they satisfy the above conditions. Therefore, at each iteration of the contour computation algorithm, each individual connected component of T, traversed by $\Gamma(v)$, can be identified.

Thus, topological changes can be detected observing the number $N_{\Gamma(v)}$ of connected subset of $\Gamma(v)$, as f evolves. We define three types of topological changes:

1. *Bifurcations*. When $N_{\Gamma(v)}$ increases from iteration t to iteration $t+1$ ($\Gamma(v)$ splits into two contours from Fig. 9a–c)
2. *Junctions*. When $N_{\Gamma(v)}$ decreases from t to $t+1$ and when several discrete contours merge (two contours merge to one from Fig. 9c–e),
3. *Terminations*. When $N_{\Gamma(v)}$ decreases from t to $t+1$, without discrete contour merge

Figure 10 shows several dual Reeb graphs obtained with this strategy, with regard to f_m (graph embedding in space will be detailed in Sect. 9).

The main contribution of our algorithm is that graph construction and simplification are performed at the same time, without input parameter or preprocessing.

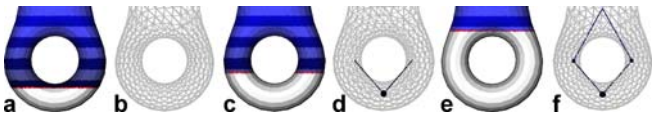


Fig. 9a–f. Bifurcation and junction contexts on a torus shape (height function)

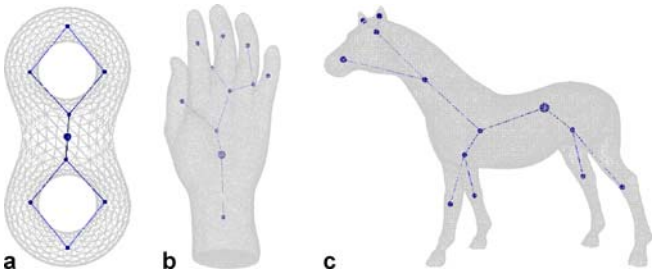


Fig. 10a–c. Dual Reeb graphs of primitive and complex shapes (f_m function)

If we compare Figs. 6 and 10, we notice that the dual Reeb graphs do not reflect the presence of noisy critical points (points in red in Fig. 6), because discrete level lines do not disconnect in those configurations.

Standard Reeb graph algorithms would have generated graphs counting as many nodes as critical points: 100 nodes for the hand model and 92 nodes for the horse model. In our approach, as contours do not disconnect in f_m non-smooth parts, only meaningful topological variations are encoded in the graph.

8 Geometrical analysis of discrete contours

Constriction approximations enable the subdivision of the branches of topological skeletons into more visually interesting parts, detecting potential articulations for example. For each *discrete contour* identified in the previous stage, its weighted curvature index is computed and local minima are identified as constrictions.

8.1 Topological constraint

Since constrictions are defined as closed curves, the analysis has to be restricted to *closed* discrete contours only.

Fig. 11a–c. Several curvature estimations on the hand model: mean curvature (**a**), Gaussian curvature (**b**) and curvature index (**c**) [19]. The curvature index generates a more important contrast between convex and concave areas

Considering each contour $\gamma(v)$ as a connected and undirected planar graph G, $\gamma(v)$ is a cycle, and consequently a closed curve, if the degree of all its vertices equals two. Therefore for each discrete contour of T reduced to a planar graph G, the degree of each of its vertices is computed and we only consider in the rest of our algorithms contours that satisfy the above property.

8.2 Contour curvature estimation

Many curvature computations have been proposed in the past [19, 21]. In this paper, surface curvature along contours is evaluated using the curvature index [19]. First, it is invariant to uniform scaling, which contributes to the stability of the algorithm. Second, experiments showed it provided more relevant results than Gaussian [32] or mean curvatures. The curvature index is computed for each vertex $v \in T$ as follows:

$$I_c(v) = \begin{cases} \frac{2}{\pi} \arctan \frac{k_1+k_2}{k_1-k_2} & \text{if } k_1 \neq k_2 \\ 0 & \text{if } k_1 = k_2 \end{cases} \quad (9)$$

with k_1 and k_2 the principal curvatures in v, computed with the algorithm described in [21]. Figure 11 shows several curvature estimations of the hand model (the mean and the Gaussian curvatures have been displayed with logarithmic scales). Notice that the curvature index generates a more important contrast between convex (in light red)

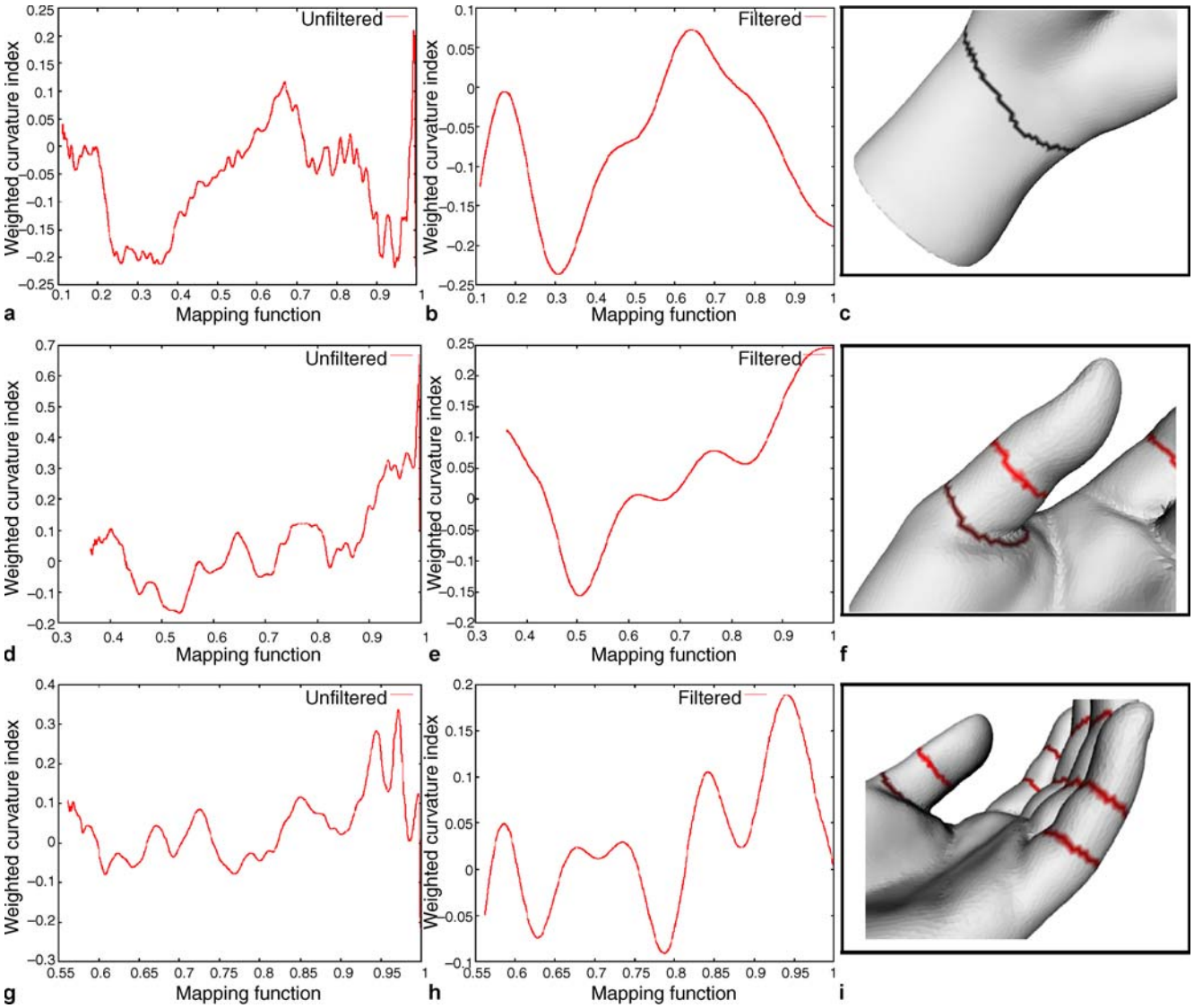


Fig. 12a–i. Unfiltered (*left column*) and filtered (*center column*, $f_\tau = 8$) curvature curves and their related components. Local negative minima have been marked with gradations of *red* on the hand model views

and concave (in dark red) areas, which benefits concave contour detection.

In order to estimate the curvature $\zeta(\gamma(v))$ associated with the discrete contour $\gamma(v)$, a weighted average of curvature indexes is computed as follows:

$$\zeta(\gamma(v)) = \frac{\sum_{\forall v_i \in \gamma(v)} I_c(v_i) \times (\mathcal{L}_{e_1}(v_i) + \mathcal{L}_{e_2}(v_i))}{2 \times \mathcal{P}(\gamma(v))}. \quad (10)$$

where $\mathcal{P}(\gamma(v))$ stands for the perimeter of the contour $\gamma(v)$ and where $\mathcal{L}_{e_1}(v_i)$ and $\mathcal{L}_{e_2}(v_i)$ stand for the lengths of the edges adjacent to v_i on the contour. Vertex curvatures are weighted so as not to give too much importance to the densely sampled parts of the contour.

8.3 Component curvature curves

During the topological analysis of discrete contours, each contour is sequentially linked to its related node in the dual Reeb graph. Thus, each node of the graph is equipped with a sorted collection of discrete contours, sorted with regard to f_m. Computing $\zeta(\gamma(v))$ for each element of this sorted collection gives a *curvature curve*, an overview of $\gamma(v)$ curvature evolution as f_m evolves.

Curves shown in Fig. 12 (left column) give examples of such evolutions on some connected components of the hand model. The f_m is reported on the x-axis while $\zeta(\gamma(v))$ is reported on the y-Axis. Left values correspond to contour curvature at the basis of components while right values correspond to contour curvature at the extremity of components.

Curvature is a well-known noise sensitive entity. Consequently, to compute nice-looking discrete contour constrictions, we have to reduce high frequency noise in curvature curves. Reducing noise on a one-dimensional data set is a trivial signal-processing problem. This can be achieved by applying an ideal low-pass filter of cutoff frequency f_τ, defined by the following transfer function:

$$H(f_{\gamma(v)}) = \begin{cases} 1 & \text{if } f_{\gamma(v)} \leq f_\tau \\ 0 & \text{if } f_{\gamma(v)} > f_\tau. \end{cases} \quad (11)$$

A filtered version of $\zeta(\gamma(v))$ is given by the following expression, where FT stands for the Fourier transform:

$$\widehat{\zeta}(\gamma(v)) = FT^{-1}(H(f_{\gamma(v)}) \times FT(\zeta(\gamma(v)))). \quad (12)$$

As shown in Fig. 12, low-pass filtering smooths curvature curves and enables a clear discrimination of local minima and maxima. The f_τ parameter has been set experimentally to 8, as discussed in Sect. 10.

8.4 Constriction selection

By definition, if $\widehat{\zeta}(\gamma(v))$ is positive, the $\gamma(v)$ neighborhood is globally convex, otherwise it is concave. Constrictions appear on the narrowest, or the most concave, parts of a surface. Consequently, local negative minima are identified as contour constrictions. In Fig. 12b, one negative minimum is identified and is marked as a constriction in Fig. 12c. Similarly, two negative minima are identified in Fig. 12e and are displayed in red in Fig. 12f.

Constriction detection is motivated by a need to segment skeleton components into more meaningful subparts. Nevertheless, to keep the high-level interest of such a refinement, *over*-segmentation must be avoided. Roughly, if two contours that are negative minima of $\widehat{\zeta}(\gamma(v))$ are too close to each other (if the difference between their average value of f_m is lower than a given threshold $\Delta = 0.1$), only the most concave one is identified as a *meaningful constriction*.

9 Topological skeleton enhancement

In previous stages of the framework, a dual Reeb graph has been provided by a topological analysis of discrete contours. This graph is composed of nodes and edges which respectively represent identified connected components and adjacency relations between them. Each node of the graph references a collection of vertices as well as the collection of corresponding discrete contours. Moreover, constriction detection has been provided by a geometrical analysis of discrete contours. Consequently, for each node of the dual Reeb graph, its contour and vertex collections can be subdivided using contour constriction as boundaries. Thus, each identified connected component of the mesh can be subdivided into more visually meaningful subparts.

Based on these decomposition processes, several embeddings of enhanced topological skeletons can be provided. For display purposes, a graph embedding (Fig. 13a) can be computed by placing a node at the Euclidean barycenter of the related mesh patch.

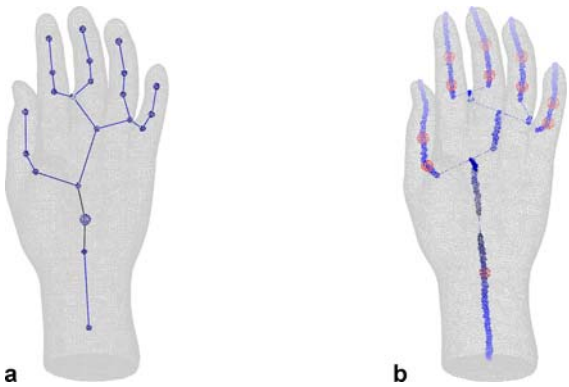


Fig. 13a,b. Enhanced topological skeleton embeddings: graph embedding (**a**) and medial axis embedding (**b**)

For some applications, it is important to constrain the skeleton position strictly inside the object. Moreover, particularly for mesh deformation, a link between vertices and skeleton branches will be additionally needed. Most of the time, this association is performed computing for each vertex its closest point on the skeleton. Thanks to their theoretical foundations, enhanced topological skeletons are appropriate candidates. Firstly, a medial axis embedding (Fig. 13b) can be computed, placing a node at the Euclidean barycenter of each discrete contour. Secondly, constriction approximation can be reported with a special node (in red in Fig. 13b), denoting a potential articulation in the object. As discrete contours have been defined relatively to a vertex, our algorithm provides a natural equivalence between the vertices of the mesh and the points of the skeleton.

10 Experiments and results

In this section, we present and comment on experimental results obtained with our method. The discussion will be laid out as follows. Firstly, experiments for parameter setting are detailed. Secondly, we focus on framework evaluation. To our knowledge, no ground-truth evaluation process has been proposed in the past for evaluating skeleton extraction accuracy. Nevertheless, we retained four evaluation criteria: time execution complexity, intrinsic properties (enumerated by Wu et al. [33]), robustness to surface degradation and usability (Sect. 11). Thirdly, framework limitations will be detailed.

Models presented in this discussion are closed connected triangulated surfaces extracted from the Princeton Shape Benchmark [29] and the INRIA Gamma Research Group [13] repositories.

10.1 Parameter setting

Feature point extraction is the first stage of the pipeline that requires a threshold value setting. This parameter (ϵ) stands for the radius of geodesic neighborhoods during the geodesic based mapping function extrema intersection process (extrema representing a same feature point might not appear on the same vertex but in the same neighborhood). If ϵ is too small, extrema will have to be very close to be merged; consequently, some feature points might be missed. On the contrary, if it is too high, distant extrema will tend to be merged into the same feature point; thus some feature points might represent several protrusions. Table 1 shows that the number of extracted feature points is quite stable when ϵ varies. The most visual perception conforming results have been obtained setting $\epsilon = 0.05$ (bold column, see Fig. 14 for graphical results).

The number of geodesic based mapping functions used for intersection (noted N_G) can also be discussed as a parameter of the approach. Table 2 shows experiments with a variant of the proposed algorithm, using a varying number of mapping functions. It shows that the more mapping functions are intersected, the less feature points are identified, as a consequence of the intersection process of multiple sets.

Discrete contour constriction detection is conditioned by the cutoff frequency f_τ of the low-pass filter applied

Table 1. Number of feature points with different ϵ parameter values

Model	$\epsilon = 0.01$	$\epsilon = 0.02$	$\boldsymbol{\epsilon = 0.05}$	$\epsilon = 0.1$	$\epsilon = 0.2$
Sphere	2	2	**2**	2	2
Bottle	2	2	**2**	2	2
Bird	9	9	**7**	7	6
Dinosaur	6	8	**7**	7	7
Horse	6	6	**7**	7	7
Hand	2	5	**6**	6	8

Table 2. Number of feature points when the number N_G of geodesic mapping functions for intersection increases. Intersecting more and more sets reduces the number of common elements

Model	$\boldsymbol{N_G = 2}$	$N_G = 3$	$N_G = 4$	$N_G = 5$	$N_G = 10$
Sphere	**2**	2	2	2	2
Bottle	**2**	2	2	2	2
Bird	**7**	6	5	5	4
Dinosaur	**7**	5	5	4	2
Horse	**7**	6	2	2	2
Hand	**6**	1	0	0	0

Table 3. Number of discrete contour constrictions with different f_τ parameter values. Most visually conforming results have been obtained with $f_\tau = 8$

Model	$f_\tau = 1$	$f_\tau = 5$	$\boldsymbol{f_\tau = 8}$	$f_\tau = 10$	$f_\tau = 20$
Sphere	0	0	**0**	0	0
Bottle	0	1	**1**	2	2
Bird	0	8	**8**	8	8
Dinosaur	0	12	**13**	15	13
Horse	0	12	**16**	14	13
Hand	0	10	**11**	13	13

Table 4. Number of discrete contour constrictions with different Δ parameter values. Setting $\Delta = 0.1$ avoids over-segmentation on concerned objects (dinosaur and horse models) without affecting constriction selection on others

Model	$\Delta = 0$	$\Delta = 0.05$	$\boldsymbol{\Delta = 0.1}$	$\Delta = 0.2$
Sphere	0	0	**0**	0
Bottle	1	1	**1**	1
Bird	8	8	**8**	8
Dinosaur	17	17	**13**	10
Horse	18	18	**16**	10
Hand	11	11	**11**	6

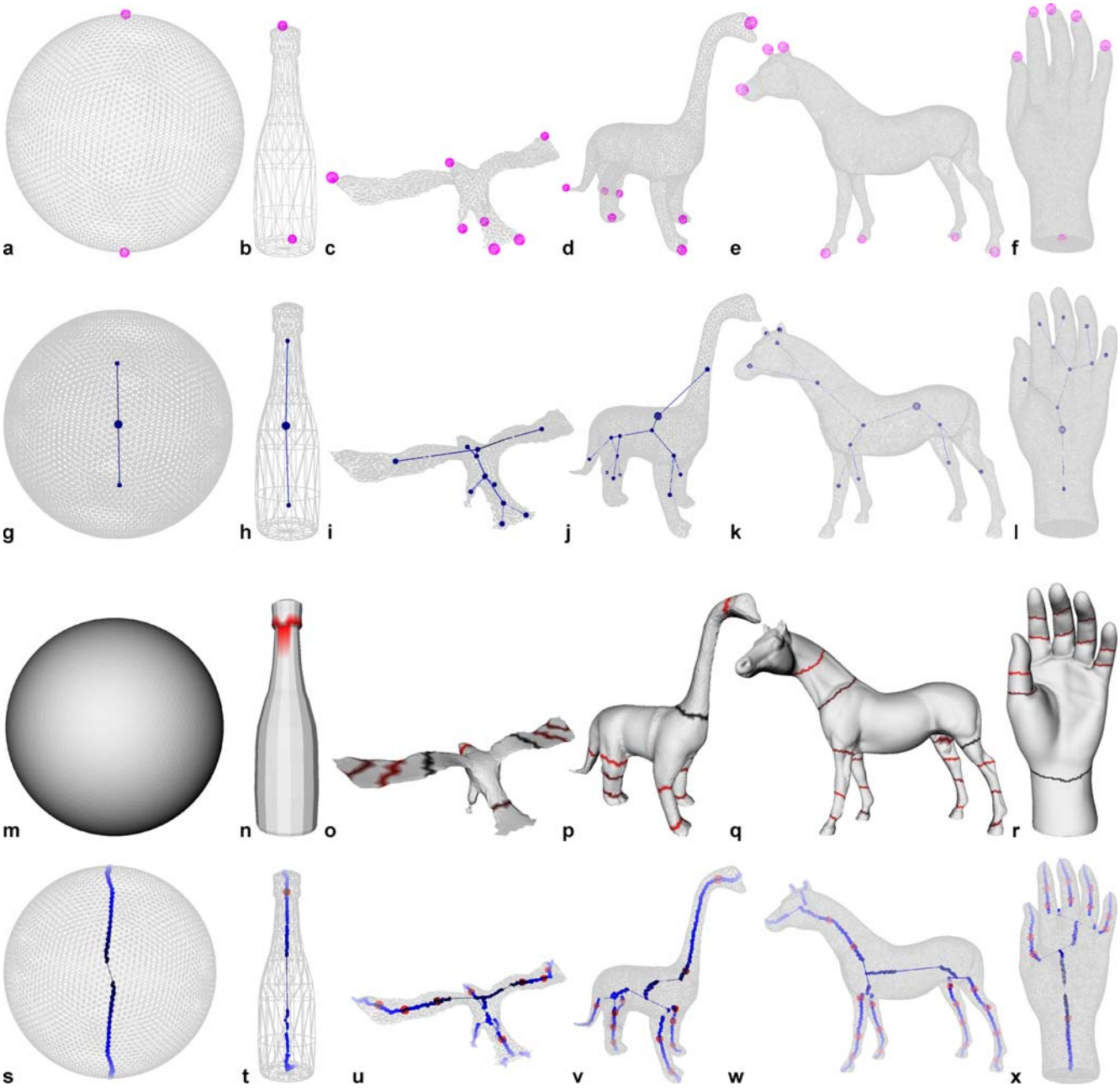


Fig. 14a–x. Feature points, dual Reeb graph, constriction approximations and enhanced topological skeleton of simple and complex objects

on curvature curves. This filter's objective is to decrease curvature high frequency noise influence. Setting $f_\tau = 8$ has shown to result in nice-looking constrictions (see Fig. 14 and Table 3). Moreover, constriction selection is also conditioned by a threshold parameter (Δ) which denotes the minimal acceptable distance between two consecutive constrictions. Table 4 shows it is not a critical parameter though it helps in avoiding over-segmentation on two models.

As a conclusion, Tables 1–4 show that most of those parameters are not critical for the stability of the overall method and that proposed values (used for Figs. 14 and 15) give visual perception conforming results.

10.2 Time complexity

Given an input closed connected triangulated surface T, let n be the number of vertices in T. Feature point ex-

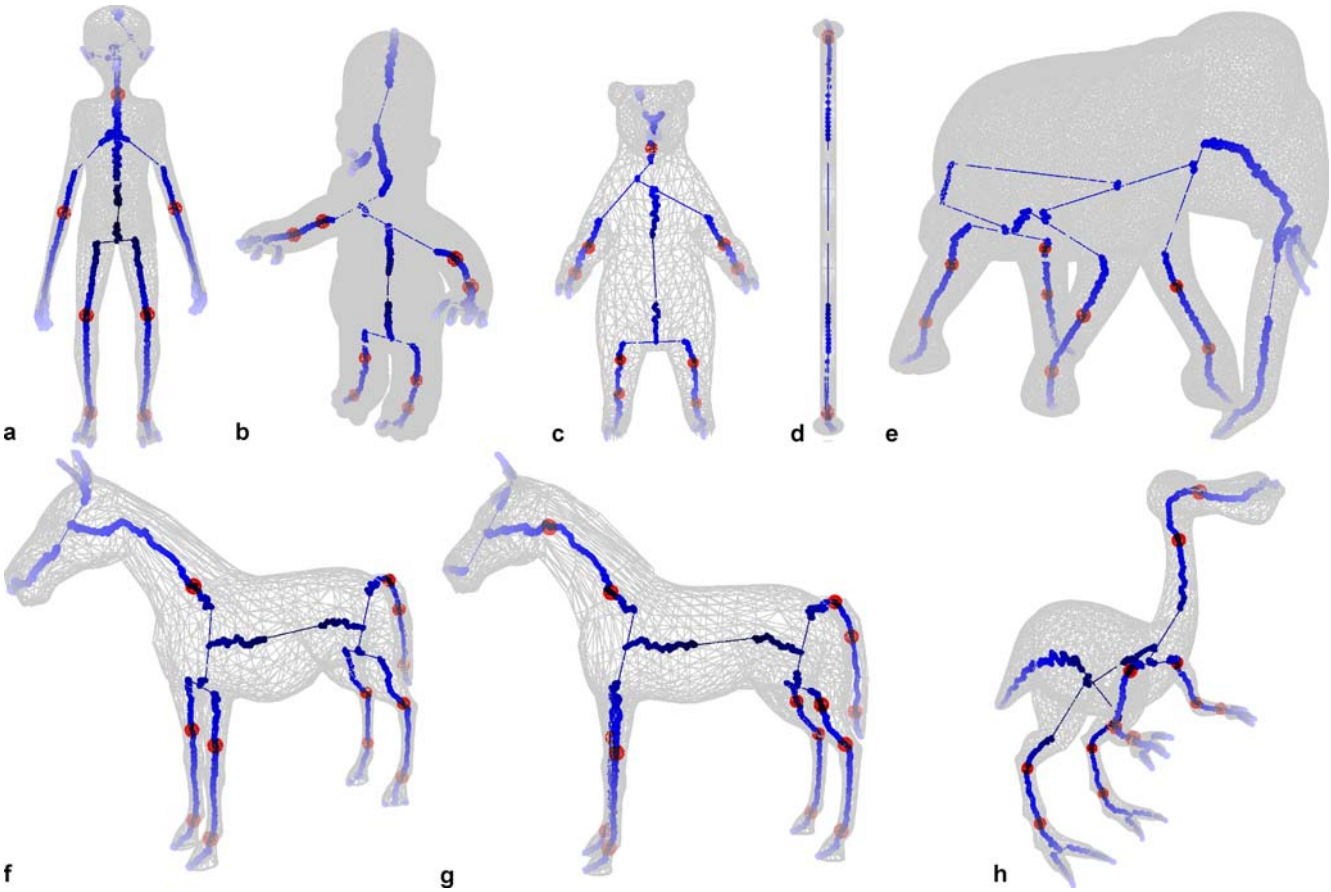


Fig. 15a–h. Enhanced topological skeleton of various objects

traction is performed in $O(n \times \log(n))$ steps. The term f_m is computed in $O(|F| \times n \times \log(n))$ steps with $|F|$ the number of identified feature points. Notice that f_m has a lower computational cost than the function proposed in [16] ($|F|$ rarely exceeds 20). Each discrete contour computation takes $O(\log(n) + n)$. Therefore, as contours are computed for each vertex in T, the overall discrete contour computation takes $O(n^2)$ steps. Topological and geometrical analyses are more straightforward. Topological analysis is performed in $O(n)$ steps. Curvature curves are computed in $O(n)$ steps. Their smoothing is realized in $O(n \times \log(n))$, using the fast Fourier transform algorithm. Consequently, we can state that the overall complexity of our method is bounded by the discrete contour computation, which takes $O(n^2)$ steps in the worst case.

Presented algorithms have been implemented in C language under GNU/Linux and experimented on a desktop PC with a 3 GHz P4-CPU and 2 gigabytes of RAM. Table 5 shows the computation times corresponding to the models presented in Fig. 14.

For comparison, the latest constriction detection algorithms [14] run in 80 seconds for a 10 000 faced model.

Table 5. Computation times

Model	Faces	Feature pts	Constrictions	Time (s)
Sphere	8000	2	0	6.342
Bottle	520	2	1	0.045
Bird	2000	7	8	0.425
Dinosaur	10 000	7	13	3.490
Horse	40 000	7	16	33.793
Hand	52 000	6	11	109.194

Domain connected graphs [33] are computed in more than 1900 seconds for such models. Three-dimensional mesh segmentation based on feature points and core extraction [17] takes 28 seconds for 4000 faced models. Notice that our overall method has a significantly lower running time than these state-of-the-art methods for equivalently sampled meshes.

10.3 Discussion

Intermediary results and topological skeletons are shown in Figs. 14 and 15. On these illustrations, the reader can notice that enhanced topological skeletons concisely en-

code both shape topology (Reeb graph, in blue) and geometry (constrictions, in red). On the one hand, the topological analysis results in the identification of the most significant topological components only, preserving the topological properties of the shape: model in Fig. 15e is a 1-genus surface and its skeleton consequently contains a cycle. On the other hand, the geometrical analysis enables the subdivision of these topological components into more visually meaningful subparts, increasing the description quality of traditional topological skeletons (subdividing humanoid models' limbs along potential articulations).

According to Wu et al. [33], a shape skeleton is expected to respect certain intrinsic properties: simplicity, stability, meaningfulness, neutrality and hierarchy. Firstly, as illustrated in Fig. 13a, enhanced topological skeletons can be collapsed into very concise representations. Secondly, the stability of the presented method can be observed in Fig. 15f and Fig. 15g, where two different horse models have nearly identical skeletons. More generally, the overall robustness of the framework will be addressed later in this discussion. Thirdly, the reader can see that no noisy details are encoded in the skeletons, which shows the meaningfulness of the framework. Fourthly, the neutrality of the skeleton is provided by the medial axis embedding we proposed in Sect. 9. Fifthly, a natural skeleton hierarchy is provided by the mapping function f_m: skeleton subparts at the center of the object, at the root of the hierarchy, have a low function value (denoted in dark blue in Figs. 14 and 15) whereas extremity parts have a high function value (denoted in light blue in Figs. 14 and 15). Consequently, we can state that enhanced topological skeletons respect the intrinsic properties of shape skeletons [33], and consequently provides a good description of 3D meshes.

Feature points have been extracted in the past. Contrary to the algorithm presented by Mortara et al. [23], the presented method resolves extraction on constant curvature areas, such as a sphere where two poles are extracted (Fig. 14a). Moreover, the algorithm presented by Katz et al. [17] runs in quadratic execution complexity while our algorithm needs $O(n \times \log(n))$ steps, with n the number of vertices in the mesh. At last, we can state that our feature point extraction algorithm leads to visual perception conforming results, as shown in Fig. 16b, c where feature points have been extracted on the extremity of each finger and toe.

As for the topological skeleton construction algorithm, the reader can notice that no noisy details have been encoded in the skeletons, because discrete contours do not disconnect in f_m non-smooth configurations (denoted with red and black critical points in Fig. 6). Contrary to traditional Reeb graph simplification algorithms, no noise-corrective threshold is required.

Discrete contour constriction detection also leads to visual perception conforming results, identifying the most significant surface bottlenecks (such as the humanoid models' elbows and knees for example). As shown in Fig. 14n, o the proposed algorithms leads to acceptable results even with coarsely designed objects, even on strongly tubular components. Notice that it also behaves correctly with primitive shapes, such as the sphere (Fig. 14m), where no constriction is identified. Moreover, contrary to previous works [14, 15], only *meaningful* constrictions are extracted.

10.4 Robustness against surface degradation

As f_m is based on normalized geodesic distance evaluation, the algorithm is invariant to geometrical transformations (translation, rotation and uniform scaling). The fol-

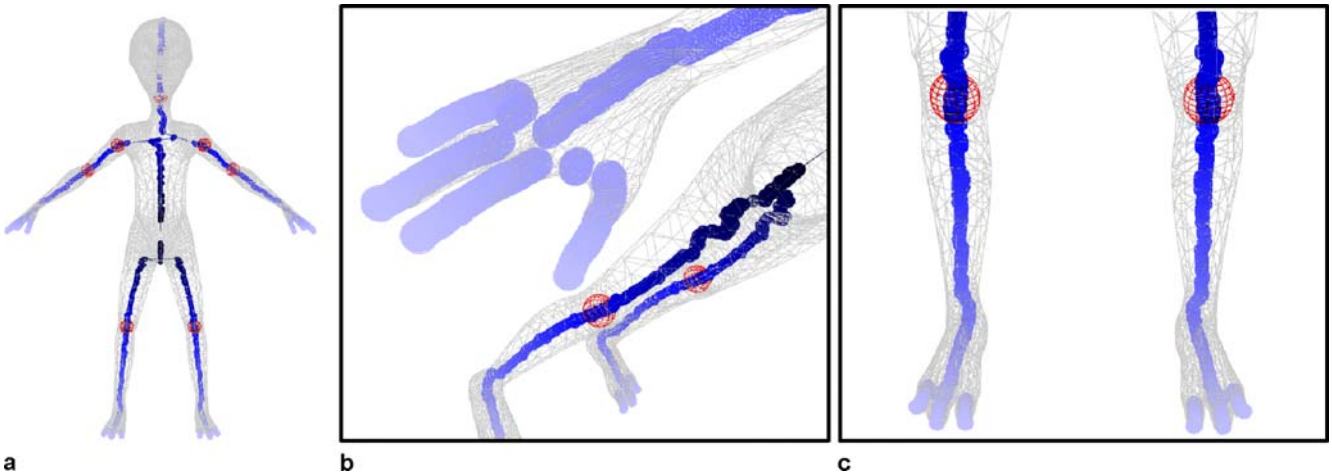


Fig. 16a–c. Zooming in the alien model, we can see that feature points have been extracted on each finger and toe, thus creating corresponding branches in the skeleton

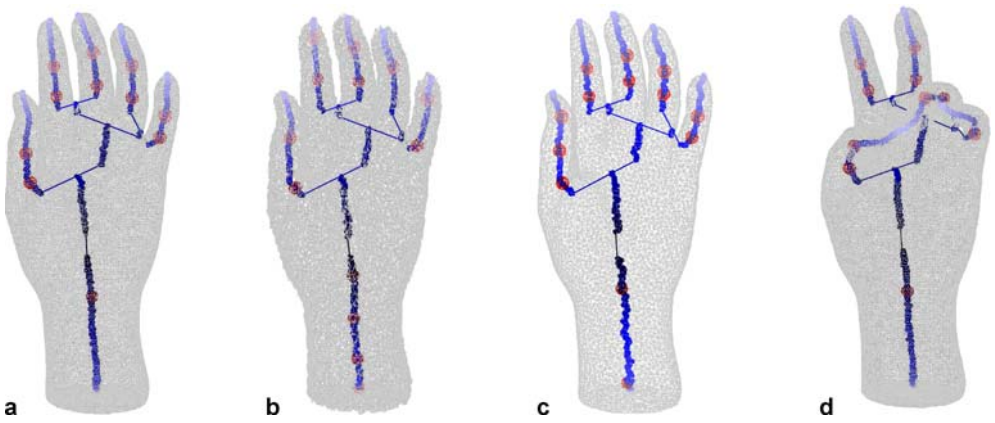


Fig. 17a–d. Algorithm robustness against surface degradations: original (**a**), surface noise (**b**), under-sampling (**c**) and user deformations (**d**)

lowing experiments try to estimate our framework's stability against more complex transformations.

The first experiment deals with surface noise (Fig. 17b), scanning acquisition noise for example. Each vertex of the input triangulation has been moved randomly in a box whose volume corresponds to 1% of the bounding box of the overall object. One can see that the topological description (in blue) remains unchanged while the geometrical description (in red) slightly varies. Notice that such a noise dramatically affects the surface and is known to be critical for medial axis transform or segmentation based methods.

Compared to [1], no re-meshing preprocess is required in our framework. To evaluate our method dependence on the triangulation of the surface, we propose in the second experiment to divide by 5 the number of vertices in the mesh, as shown in Fig. 17c. In a similar way to the previous experiment, surface under-sampling slightly affects the constriction detection process only.

The third experiment's purpose is to evaluate enhanced topological skeletons robustness against variations in model pose. This property benefits applications such as surface comparison. Geodesic metrics are commonly attributed to the property to be invariant to variations in model pose though it is rarely used. In Fig. 17d, the hand model has been deformed using rigid rotations (as detailed in Sect. 11), and the enhanced topological skeleton of the deformed surface has been computed. Notice not only the topological but also the geometrical analyses behaved nearly identically: only one constriction is missing in the little finger and the other ones kept precisely their original position. Thus, we can conclude from this experiment that the presented method is also robust against variations in model pose.

10.5 Framework limitations

In Fig. 15c, the bear model seems symmetrical while it is not rigorous. This results in a skeleton whose structure is not symmetrical. Generally speaking, topological skeletons (because of their theoretical foundations) tend to amplify the models' asymmetry. We believe that for future work, capturing the notion of *perceived symmetry* could increase the descriptive quality of the framework, inserting in an artificial way some symmetry in the skeleton if needed. In this sense, recent results in geometry processing for symmetry detection [27] are worth investigating. This property could benefit applications such as skeleton based surface similarity estimation, which often uses asymmetry sensitive graph-matching algorithms.

Moreover, the proposed method assumes that constrictions appear along extracted discrete contours. Even if it is verified in practice for most tested objects (see Figs. 14 and 15), this is a strong hypothesis. An undesired consequence of this hypothesis is presented in Fig. 18, where a contour passing through a deeply concave area is incorrectly identified as a constriction. Reciprocally, expected

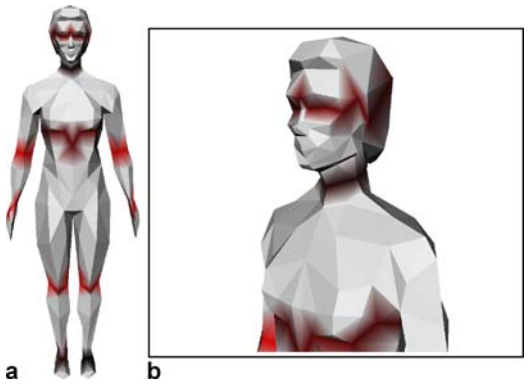


Fig. 18a,b. False positive constriction on the head of a coarsely designed humanoid model. As the eye areas are deeply concave, the associated discrete contour is incorrectly identified as a constriction

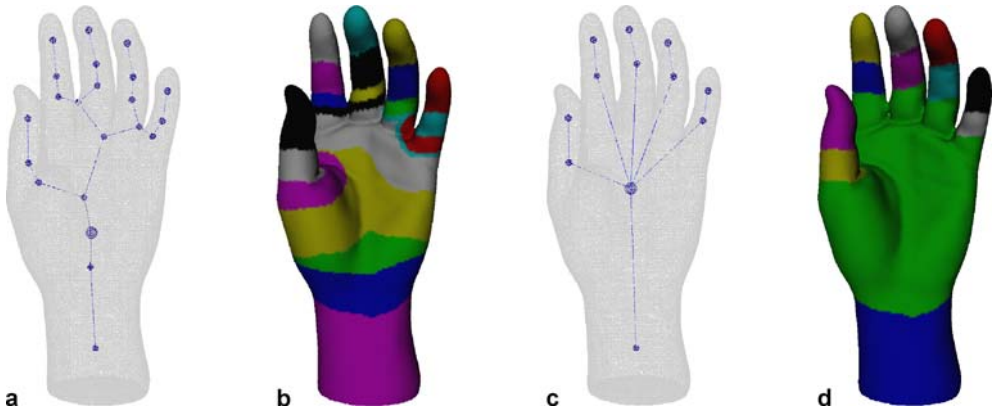


Fig. 19a–d. Topology driven segmentation overview: a raw segmentation (**b**) is obtained from the enhanced topological skeleton (**a**), then a node merging algorithm (**c**) completes the process (**d**)

constrictions that are not located along discrete contours will not be identified. Graph-cut based optimization algorithms could be a solution to this problem, moving constriction approximations to their exact location, helping the constriction selection.

11 Examples of application

In previous section, our framework has been evaluated according to its time execution complexity, its descriptive properties and its robustness against surface degradation. In this section, we focus on its application as its last evaluation criteria and propose two applications: topology driven mesh segmentation and skeleton driven mesh deformation.

11.1 Topology driven mesh segmentation

Mesh segmentation roughly consists in cutting a surface into meaningful subparts [2]. Similarly to skeleton extraction, it benefits a large spectrum of applications. Recently, the need for *compatible segmentation*, meaning segmenting identically two meshes representing the same class of object, has been expressed [17]. Contrary to geometrical low-level based approaches [34], a solution to this issue could reside in driving the segmentation using high-level shape information, such as symmetry or topology. In this sense, Berreti et al. [3] proposed the use of the topological description provided in [16] to compute simple, but similar segmentations. Here, we propose to exploit this idea and to take advantage of the higher descriptive quality of our skeletons (in comparison to traditional topological skeletons) to improve topology driven mesh segmentation.

The proposed algorithm is summed up in Fig. 19. First, for each node of the graph, its related surface patch is displayed with a distinctive color. This results in a raw oversegmentation (Fig. 19b). Then, the graph is simplified (Fig. 19c) and the final segmentation (Fig. 19d) is computed using the following two node-merging heuristics:

1. *Delete thin patches*. If the degree of processed node is greater than 2, then merge it with its adjacent node of biggest area, except if a constriction separates them.
2. *Select the most concave boundaries*. If the degree of processed node is equal to 2 and it is the first node of a branch and its last contour has a $\widehat{\zeta}(\gamma(v))$ value lower than a given threshold $\widehat{\zeta}_{\min}$ (fixed to -0.1), then merge it with its parent node (its adjacent node of lowest average f_m value).

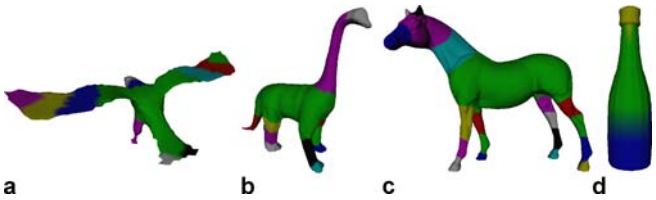


Fig. 20a–d. Topology driven segmentation results for some test objects. The proposed algorithm identifies core and limbs, subdividing limbs along deeply concave areas

More results are shown in Fig. 20. In these illustrations, objects are segmented into core and limbs and limbs are segmented along constrictions. Notice that this algorithm benefits our skeleton extraction framework properties, and particularly its robustness against surface degradation.

11.2 Skeleton driven mesh deformation

User defined 3D model deformations often need tricky mesh editing operations. Shape skeletons have shown to

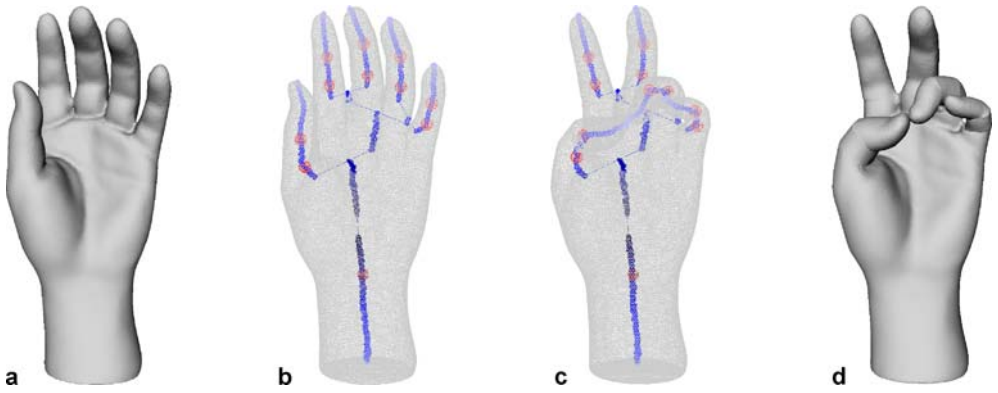


Fig. 21a–d. Application to mesh deformation. The user directly grabs the branches of the skeleton (**b**) and rotates them using constrictions (in *red*) as articulations (**c**). Thanks to the equivalence relation between the vertices of the mesh and the points of the skeleton, deformations are directly reflected on the object (**d**)

be user-friendly shape abstractions for surface deformation [8, 18, 33]. Basically, the user grabs the branches of the skeleton and moves them to produce the desired deformation. Contrary to conventional topological skeletons, our skeletons are enhanced with constrictions which mostly corresponds to articulations. Consequently, the user can employ these special points of the skeleton (in red in Fig. 21) as natural rotation reference points. Once a branch of the skeleton has been rotated, the corresponding rotation matrix is constructed. Then, for each vertex of the related surface patch, its new position is computed multiplying its original position vector by the rotation matrix. Figure 21 shows that nice-looking deformations can be obtained with only a few user interactions, thanks to the high-level description provided by enhanced topological skeletons.

12 Conclusion and future work

In this paper, we presented a new concise shape abstraction, *enhanced topological skeleton*, that encodes both topological and geometrical evolutions of some mapping function contours. We also proposed a unified and fully automatic algorithm for its extraction. First, it computes a Reeb graph. Then it refines it by using discrete contour constriction as boundaries. To the authors' knowledge, this is the first approach that unifies Reeb graph construction and constriction computations.

Three scientific contributions are given. First, we proposed a robust and straightforward algorithm for feature point extraction. It enables the computation of an invariant mapping function which reveals well the shape most significant features. Secondly, we presented an algorithm for *discrete contour* computation. We showed that a topological analysis of these discrete contours enables a unified Reeb graph construction and simplification process. Finally, we showed that a geometrical analysis of the discrete contours provides visual perception conforming constriction approximations, enabling the refinement of the traditional Reeb graphs into more visually meaningful skeletons.

Moreover, we presented an extensive evaluation discussion of experimental results, according to four criteria. First, we have shown that *enhanced topological skeletons* respect shape skeleton intrinsic descriptive properties [33], improving traditional topological skeletons description quality. Second, we have shown that the presented algorithm runs significantly faster than state-of-the-art methods. Third, we have shown that enhanced topological skeletons are affine-invariant and robust to a variety of surface degradations, such as surface noise, mesh sampling variation and surface deformation. And finally, to evaluate the usability of the presented framework, we developed two applications related to topology driven mesh segmentation and skeleton driven mesh deformation.

Some limitations have been enumerated in the discussion. In future work, we would like to capture the notion of *perceived symmetry* in order to extract symmetrical skeletons if the objects seem roughly symmetrical. We believe this could benefit many applications, like surface similarity estimation. As a long term perspective, we would like to use enhanced topological skeletons as a shape abstraction for mesh editing operations within the framework of *modeling by example* systems [12].

Acknowledgement The authors would like to thank Pacific Graphics anonymous reviewers for their relevant comments, which helped to improve the framework. This work is partially supported by the European Network of Excellence *Delos* No. 507618 (http://www.delos.info).

References

1. Attene, M., Biasotti, S., Spagnuolo, M.: Shape understanding by contour-driven retiling. Vis. Comput. **19**, 127–138 (2003)
2. Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., Tal, A.: Mesh segmentation: A comparative study. In: Shape Modeling International, pp. 14–25. IEEE Computer Society, Matsushima, Japan (2006)
3. Berreti, S., Del Bimbo, A., Pala, P.: Partitioning of 3D meshes using Reeb graphs. In: IEEE ICPR, pp. 19–22. IEEE Computer Society, Hong Kong (2006)
4. Biasotti, S., Marini, S., Mortara, M., Patanè, G.: An overview on properties and efficacy of topological skeletons in shape modelling. In: Shape Modeling International, pp. 245–254. IEEE Computer Society, Seoul, Korea (2003)
5. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychol. Rev. **94**, 115–147 (1987)
6. Blum, H., Nagel, R.N.: Shape description using weighted symmetric axis features. Pattern Recognit. **10**, 167–180 (1978)
7. Bremer, P.T., Edelsbrunner, H., Hamann, B., Pascucci, V.: Topological hierarchy for functions on triangulated surfaces. IEEE Trans. Vis. Comput. Graph. **10**, 385–396 (2004)
8. Capell, S., Green Seth abd Curless, B., Duchamp, T., Popović, Z.: Interactive skeleton-driven dynamic deformations. ACM Trans. Graph. **21**, 586–593 (2002)
9. Carr, H., Snoeyink, J., de Panne, M.V.: Simplifying flexible isosurfaces using local geometric measures. In: IEEE Visualization, pp. 497–504. IEEE Computer Society, Minneapolis, USA (2004)
10. Cole-McLaughlin, K., Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Loops in Reeb graphs of 2-manifolds. In: Symposium on Computational Geometry, pp. 344–350. ACM, San Diego, USA (2003)
11. Edelsbrunner, H., Mücke, E.P.: Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. ACM Trans. Graph. **9**, 66–104 (1990)
12. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. ACM Trans. Graph. **23**, 652–663 (2004)
13. Gamma Research Group Repository: INRIA. http://www-c.inria.fr/gamma/disclaimer.php (2006). Accessed 2006
14. Hétroy, F.: Constriction computation using surface curvature. In: Eurographics, pp. 1–4. Eurographics Association, Dublin, Ireland (2005)
15. Hétroy, F., Attali, D.: From a closed piecewise geodesic to a constriction on a closed triangulated surface. In: Pacific Graphics, pp. 394–398. IEEE Computer Society, Canmore, Canada (2003)
16. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH, pp. 203–212. ACM Press, Los Angeles, USA (2001)
17. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. Vis. Comput. **21**, 865–875 (2005)
18. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. Graph. **22**, 954–961 (2003)
19. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. Image Vision Comput. **10**, 557–565 (1992)
20. Lazarus, F., Verroust, A.: Level set diagrams of polyhedral objects. Tech. Rep. 3546, Institut National de Recherche en Informatique et en Automatique (INRIA) (1999)
21. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. In: Visualization and Mathematics, pp. 33–57. Springer, Berlin, Germany (2002)
22. Morse, M.: Relations between the critical points of a real function of n independant variables. Trans. Am. Math. Soc. **27**, 345–396 (1925)
23. Mortara, M., Patanè, G.: Affine-invariant skeleton of 3D shapes. In: Shape Modeling International, pp. 245–252. IEEE Computer Society, Banff, Canada (2002)
24. Ni, X., Garland, M., Hart, J.: Fair Morse functions for extracting the topological structure of a surface mesh. ACM Trans. Graph. **23**, 613–622 (2004)
25. Nieda, T., Pasko, A., Kunii, T.L.: Detection and classification of topological evolution for linear metamorphosis. Vis. Comput. **22**, 346–356 (2006)
26. Ogniewicz, R., Ilg, M.: Voronoi skeletons: Theory and applications. In: IEEE Computer Vision and Pattern Recognition, pp. 63–69. IEEE Computer Society, Champaign, USA (1992)
27. Podolak, J., Shilane, P., Golovinskiy, A., Rusinkiewicz, S., Funkhouser, T.: A planar-reflective symmetry transform for 3D shapes. ACM Trans. Graph. **25**, 549–559 (2006)
28. Reeb, G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. C.R. Acad. Sci. **222**, 847–849 (1946)
29. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton shape benchmark. In: Shape Modeling International, pp. 167–178. IEEE Computer Society, Genova, Italy (2004)
30. Shinagawa, Y., Kunii, T.L., Kergosien, Y.L.: Surface coding based on Morse theory. IEEE Comput. Graph. Appl. **11**, 66–78 (1991)
31. Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T.L., Ueda, M.: Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. Comput. Graph. Forum **14**, 181–192 (1995)
32. Tierny, J., Vandeborre, J.P., Daoudi, M.: 3D mesh skeleton extraction using topological and geometrical analyses. In: Pacific Graphics, pp. 85–94. IEEE Computer Society, Taipei, Taiwan (2006)
33. Wu, F.C., Ma, W.C., Liang, R.H., Chen, B.Y., Ouhyoung, M.: Domain connected graph: the skeleton of a closed 3D shape for animation. Vis. Comput. **22**, 117–135 (2006)
34. Yamauchi, H., Gumhold, S., Zayer, R., Seidel, H.P.: Mesh segmentation driven by Gaussian curvature. Vis. Comput. **21**, 649–658 (2005)
35. Zhang, E., Mischaikow, K., Turk, G.: Feature-based surface parametrization and texture mapping. ACM Trans. Graph. **24**, 1–27 (2005)

JULIEN TIERNY received the M.Sc. degree (summa cum laude) in Computer Science from the University of Lille (USTL) along with an engineering degree from TELECOM Lille 1 in 2005. He is currently a Ph.D. candidate within the Computer Science Laboratory of the University of Lille (UMR USTL/CNRS 8022). He is also a teaching assistant in the Computer Science Department of the University of Lille. His research interests include shape modeling, shape similarity estimation, geometry processing and their applications.

JEAN-PHILIPPE VANDEBORRE received the M.S. degree in 1997 and Ph.D. degree in Computer Science in 2002, both from the University of Lille (UTSL), France. Currently, he is an associate professor at TELECOM Lille 1 (a graduate engineering school) in the Computer Science and Network Department. He is also a member of the Computer Science Laboratory at the University of Lille (UMR USTL/CNRS 8022). His current research interests are mainly focused on three-dimensional model analysis, and include multimedia indexing and retrieval from content and their applications.

MOHAMED DAOUDI is a Full Professor of Computer Science in the GET/TELECOM Lille 1. He received the Ph.D. degree in Computer Engineering from the University of Lille 1 (USTL), France, in 1993 and Habilitation á Dirger des Recherches (HDR) from the University of Littoral, France, in 2000. He was the head of the MIIRE research group of LIFL (UMR USTL-CRNS 8022) from 2000–2004. His research interests include pattern recognition, image processing, invariant representation of images and shapes, three-dimensional analysis and retrieval and more recently 3D face recognition.
He has published more than 80 papers in refereed journals and proceedings of international conferences.
He has served as a Program Commitee member for the International Conference on Pattern Recognition (ICPR) in 2004 and the International Conference on Multimedia and Expo (ICME) in 2004 and 2005. He is a frequent reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence and for Pattern Recognition Letters. His research has been funded by ANR, RNRT and European Commission grants.

Annexe C

Partial 3D shape retrieval by Reeb pattern unfolding. Julien Tierny, Jean-Philippe Vandeborre, and Mohamed Daoudi. *Computer Graphics Forum*, 28(1) :41–55, March 2009.

Abstract. This paper presents a novel approach for fast and efficient partial shape retrieval on a collection of 3D shapes. Each shape is represented by a Reeb graph associated with geometrical signatures. Partial similarity between two shapes is evaluated by computing a variant of their maximum common sub-graph. By investigating Reeb graph theory, we take advantage of its intrinsic properties at two levels. First, we show that the segmentation of a shape by a Reeb graph provides charts with disk or annulus topology only. This topology control enables the computation of concise and efficient sub-part geometrical signatures based on parameterisation techniques. Secondly, we introduce the notion of Reeb pattern on a Reeb graph along with its structural signature. We show this information discards Reeb graph structural distortion and still depicts the topology of the related sub-parts. The number of combinations to evaluate in the matching process is then dramatically reduced by only considering the combinations of topology equivalent Reeb patterns. The proposed framework is invariant against rigid transformations and robust against non-rigid transformations and surface noise. It queries the collection in interactive time (from 4 to 30 seconds for the largest queries). It outperforms the competing methods of the SHREC 2007 contest in term of NDCG vector and provides, respectively, a gain of 14.1% and 40.9% on the approaches by Biasotti et al. (2006) and Cornea et al. (2005). As an application, we present an intelligent modelling-by-example system which enables a novice user to rapidly create new 3D shapes by composing shapes of a collection having similar sub-parts.

COMPUTER GRAPHICS forum

DOI: 10.1111/j.1467-8659.2008.01190.x

COMPUTER GRAPHICS forum
Volume 28 (2009), number 1 pp. 41–55

Partial 3D Shape Retrieval by Reeb Pattern Unfolding

Julien Tierny[1], Jean-Philippe Vandeborre[1,2] and Mohamed Daoudi[1,2]

[1]LIFL (UMR USTL/CNRS 8022), Lille, France
[2]Institut TELECOM ; TELECOM Lille 1, France
{julien.tierny, jean-philippe.vandeborre, mohamed.daoudi}@lifl.fr

Abstract

This paper presents a novel approach for fast and efficient partial shape retrieval on a collection of 3D shapes. Each shape is represented by a Reeb graph associated with geometrical signatures. Partial similarity between two shapes is evaluated by computing a variant of their maximum common sub-graph.

By investigating Reeb graph theory, we take advantage of its intrinsic properties at two levels. First, we show that the segmentation of a shape by a Reeb graph provides charts with disk or annulus topology only. This topology control enables the computation of concise and efficient sub-part geometrical signatures based on parameterisation techniques. Secondly, we introduce the notion of Reeb pattern *on a Reeb graph along with its structural signature. We show this information discards Reeb graph structural distortion and still depicts the topology of the related sub-parts. The number of combinations to evaluate in the matching process is then dramatically reduced by only considering the combinations of topology equivalent Reeb patterns.*

The proposed framework is invariant against rigid transformations and robust against non-rigid transformations and surface noise. It queries the collection in interactive time (from 4 to 30 seconds for the largest queries). It outperforms the competing methods of the SHREC 2007 contest in term of NDCG vector and provides, respectively, a gain of 14.1% and 40.9% on the approaches by Biasotti et al. [*BMSF06*] *and Cornea et al.* [*CDS*05*].

As an application, we present an intelligent modelling-by-example system which enables a novice user to rapidly create new 3D shapes by composing shapes of a collection having similar sub-parts.

Keywords: 3D shape retrieval, reeb graph theory, partial matching, shape signature, modeling-by-example

ACM CCS: H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing I.3.5 [Computer graphics]: Computational Geometry and Object Modelling

1. Introduction

Three dimensional (3D) shape retrieval systems based on visual similarity aim at helping human users browsing large collections of 3D shapes in an interactive and intuitive way. In this framework, the user specifies a 3D model as an example query and the system is expected to sort the entries of the collection by decreasing visual similarity, providing as top results the most similar entries.

An important literature has been provided for shape retrieval based on global similarity, presenting methods enabling retrieval of similar objects despite rigid transformations [CTSO03, FMK*03] or even non-rigid transformations (such as shape bending or character articulation) [HSKK01]; [GSCO07]; [JZ07]. We defer the reader to survey articles [TV04, BKS*05, IJL*05] for a broad overview of retrieval methods based on global similarity.

However, other applicative contexts, such as modelling-by-example [FKS*04] (where new shapes are created by cutting and pasting existing shape sub-parts) or classification [HKDH04], might require to retrieve objects with regard to *partial* similarity. In such a paradigm, systems are expected

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Submitted February 2008
Revised April 2008
Accepted July 2008

to retrieve objects that have *similar sub-parts* even if they visually differ globally.

Unlike shape retrieval based on global similarity, only few papers have addressed the partial shape retrieval problem, while it is seen as the next challenging open issue by the shape retrieval community [CTSO03, FMK*03]. Moreover, it is a more general problem than global shape similarity estimation (two globally similar shapes will also be similar partially) and thus it is a research topic of larger impact.

Partial shape retrieval state-of-the-art techniques can be roughly classified into two categories.

On the one hand, local descriptors-based techniques aim at characterizing the local properties of a large number of small features extracted on the shape. Then, partial shape similarity is estimated by feature point-to-point matching and matching similarity estimation. Liu *et al.* [LZQ06] propose to use a Monte-Carlo sampling on the surface model and to capture the local aspect of the shape with spin-image signatures [JH99]. Funkhouser and Shilane [FS06] present a more sophisticated sampling strategy and then describe local geometry with several descriptors based on Spherical Harmonics [FMK*03] but this method is only experimented for global shape retrieval. In order to deal with the combinatorial explosion due to individual local feature comparison, such approaches use complex data structures to drive the feature matching process, such as priority queues [FS06] or feature clusters [LZQ06] inspired by text-document analysis. Gal *et al.* [GCO06] present an interesting geometrical hashing mechanism associated with a local surface description based on curvature analysis. In the context of Euclidean partial self symmetry detection, Mitra *et al.* [MGP06] use a similar approach by sampling the surface and associating samples with geometrical signatures based on normal cycles [CSM03]. Then, the authors propose a sample pruning strategy specific to symmetry characterization. However, as underlined by Biasotti *et al.* [BMSF06], most of local descriptor methods base their partial similarity estimation on point-to-point matching only. This is particularly detrimental in term of re-usability in application contexts such as modelling-by-example [FKS*04], where the similar sub-parts have to be explicitly identified and extracted.

On the other hand, structural-based approaches present the advantage to explicitly identify the surface patches that have been matched. Moreover, the combinatorial explosion due to feature comparison can be easily reduced by reasoning on the structure of the shape instead of using complex hashing mechanisms. These methods first segment the shape and represent it by a graph (or a skeleton [DJ06]) depicting the structural relations between the segments. Then, partial shape similarity is estimated using graph matching techniques. Cornea *et al.* [CDS*05] propose to extract the curve-skeleton of the shape and then employ the Earth Mover's Distance [KSDD03] to evaluate the partial similarity of the skeletons. However, sub-part signature is based on the euclidean distance between the surface and its skeleton which makes the method quite sensitive to non-rigid transformations. Biasotti *et al.* [BMSF06] present an efficient method based on Reeb graphs and Spherical Harmonics [FMK*03]. However, even if the Reeb graph construction algorithm is robust to non-rigid transformations (such as shape bending or character articulation), sub-parts signatures (spherical harmonics) are not, which is slightly detrimental to the overall robustness of the approach.

In this paper, we present a fast and efficient structural-oriented approach for partial 3D shape retrieval, based on Reeb graphs—an expressive topological shape description which has already shown its utility in several computer graphics applications [BMS00, ZMT05, AHLD07] (see [BGSF08] for a recent survey). Unlike previous works based on Reeb graphs [HSKK01, TS05] and which have been specifically designed for global shape retrieval (using a multi-resolution strategy), this work presents a framework for *partial* similarity retrieval (retrieving objects that have similar sub-parts even if the visually differ globally). Moreover, in comparison with the technique proposed by Biasotti *et al.* [BMSF06], this paper introduces a novel description (which is pose insensitive) and a new partial comparison method. The main contribution of this work is to take advantage of the intrinsic properties of Reeb graph theory to improve both the shape description and comparison processes. First we segment the shape using a Reeb graph and encode the relations between resulting patches into a dual Reeb graph. We show this segmentation provides only two types of charts, called *Reeb charts*, that have either disk or annulus topology. This topology control enables efficient pose insensitive sub-part geometrical description based on parameterisation techniques [WWJ*06, TVD07]. Secondly, we introduce the notion of *Reeb pattern* on a Reeb graph along with its structural signatures. We show this information discards Reeb graph structural distortion and avoids the use of error-tolerant graph matching algorithms [MB98]. Moreover, *Reeb pattern* structural signatures concisely encode the related sub-part topology and enable to dramatically reduce the number of combinations to evaluate in the matching process by only considering the combinations of topology equivalent *Reeb patterns*.

After a brief method overview, we describe the Reeb graph segmentation process and detail the sub-part geometrical signature computation. Then, we introduce the notion of Reeb pattern and present the partial similarity estimation algorithm based on Reeb pattern unfolding (RPU). Finally, we show experimental results that demonstrate the performance improvement and the robustness of our method. We also describe an intelligent modelling-by-example system that demonstrates the applicative interest of our framework.

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

2. Method Overview

Let M be a closed 2-manifold of arbitrary genus embedded in $\mathbb{R}^3$. First, we compute the Reeb graph of M to segment it into a set of charts C_i of controlled topology, which we call *Reeb charts*, which have either disk or annulus topology. Then, for each chart, we compute its unfolding signature λ_{ϕ_i} (as introduced in [TVD07]) by measuring the distortion of its mapping ϕ_i to the canonical planar domain D (either the unit disk or the unit annulus). Finally, the input manifold is represented by a dual Reeb graph noted $\mathcal{R}$ associated with the unfolding signatures of the Reeb charts.

To measure the partial similarity between two dual Reeb graphs $\mathcal{R}_1$ and $\mathcal{R}_2$, approximations of their maximally similar common sub-graphs $\mathcal{G}_1^*$ and $\mathcal{G}_2^*$ are computed. In particular, to reduce the number of combinations to consider, we use the following strategy. First, the sets $\mathcal{P}_1$ and $\mathcal{P}_2$ of topology equivalent *Reeb patterns* belonging, respectively, to $\mathcal{R}_1$ and $\mathcal{R}_2$ are detected. Then, the set $\mathcal{M}$ of all possible mappings m between topology equivalent *Reeb patterns* of $\mathcal{P}_1$ and $\mathcal{P}_2$($m : \mathcal{P}_1 \rightarrow \mathcal{P}_2$) is computed. In practice, the average cardinality of $\mathcal{M}$ is 20. For each mapping m, an expansion algorithm simultaneously expands the common sub-graphs $\mathcal{G}_1^m$ and $\mathcal{G}_2^m$ in $\mathcal{R}_1$ and $\mathcal{R}_2$.

Each mapping m is scored relatively to the geometrical similarity $S(m)$ between the two common sub-graphs $\mathcal{G}_1^m$ and $\mathcal{G}_2^m$.$S(m)$ is computed by comparing the unfolding signatures λ_{ϕ_i} and λ_{ϕ_j} of the pairs of matched Reeb charts C_i and C_j. Let $\widehat{m^*}$ maximize on $\mathcal{M}$ the similarity S between its related sub-graphs $\widehat{\mathcal{G}_1^*}$ and $\widehat{\mathcal{G}_2^*}$. The similar sub-parts of two closed 2-manifolds M_1 and M_2 are identified by $\widehat{\mathcal{G}_1^*}$ and $\widehat{\mathcal{G}_2^*}$. Moreover, the partial similarity between M_1 and M_2 is given by $S(\widehat{m^*})$. Algorithm 1 summarizes the overall process.

Algorithm 1 Partial shape retrieval algorithm overview.

```
Compute the query shape's Reeb graph R_q
for all C_i ∈ R_q do
   Compute C_i unfolding signature.
end For
Compute R_q structural signatures.
for all R_c in the collection index do
   Compute the set M of mappings m between
   topology equivalent Reeb patterns of R_q and R_c.
   for all m ∈ M do
      Expand recursively m to patterns' neighbour charts.
      Evaluate S(m).
   end for
   Return S(m*) as the partial
      similarity between R_q and R_c
end for
Sort the collection entries according to their partial
   similarity to the query shape.
```

3. Reeb Graph and Unfolding Signatures

Given an input closed 2-manifold M of arbitrary genus, the first step of the framework, the shape description step, consists in segmenting M using its Reeb graph [Ree46] and to compute a geometrical signature for each extracted surface segment.

3.1. Reeb graph segmentation

Definition 1 (Reeb graph) Let $f : M \rightarrow \mathbb{R}$ be a simple Morse function defined on a compact manifold M. The Reeb graph $R(f)$ is the quotient space on $M \times \mathbb{R}$ by the equivalence relation $(p_1, f(p_1)) \sim (p_2, f(p_2))$, which holds if $f(p_1) = f(p_2)$ and p_1, p_2 belong to the same connected component of $f^{-1}(f(p_1))$.

To deal with invariance to rigid transformations and robustness to non-rigid ones, we compute the Reeb graph of the input 2-manifold (represented by a triangulation noted T) using a function based on geodesic distances. In particular, to introduce some *visual semantics* in the segmentation, we automatically extract feature points (vertices located on the extremity of prominent components, see implementation details in Section 3.3). For each vertex $v \in T$, $f(v) = \delta(v, v_f)$ where δ stands for the geodesic distance and v_f for the closest feature point from v. Figure 2(a) shows the level lines of f and the feature points of T (in green). This computation of the f function has been preferred to the integral geodesic approximation [HSKK01] or a harmonic function [NGH04] because it gives better emphasis on local shape features (like fingers or ears) thanks to the feature points, which is crucial for partial matching.

Definition 2 (Reeb chart) Let $\Psi: M \rightarrow R(f)$ map each point p of M to its equivalence class in $R(f)$. Let $E = \{E_0, \ldots E_n\}$ be the edges (maximally connected unions of equivalence classes containing only regular points of f) of the Reeb graph $R(f)$. $C_i = \Psi^{-1}(E_i)$ is defined as a *Reeb chart*.

Figure 2(b) shows a dual Reeb graph (where each edge E_i is collapsed in a coloured node and where black arcs represent adjacency relations between edges). Figure 1 gives more example of feature points, f functions and dual Reeb graphs. Figure 2(c) and 2(d) show the segmentation of the hand model into its Reeb charts. Basically, Reeb charts are the surface patches that correspond to the nodes of the dual Reeb graph.

Statement 1 (Reeb chart topology) Reeb charts of a compact closed orientable 2-manifold have either disk or annulus topology whatever the genus of the manifold is.

This statement can be briefly argued as follows. By definition, an edge E_i has two extremities, whose pre-images by Ψ are circles which form the two boundary components of the closure of the chart C_i. Hence, the closure of the Reeb charts has two boundary components. Morever, C_i has genus

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

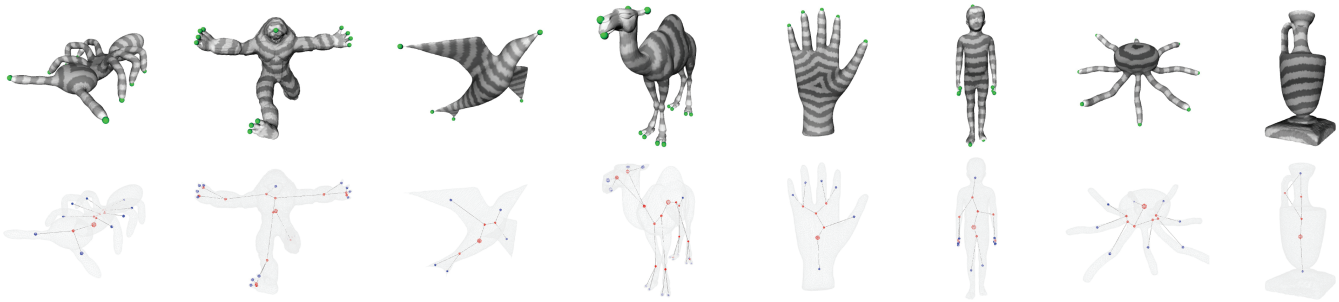


Figure 1: *Feature points and f functions (first row) and dual Reeb graphs (second row) of several surface models.*

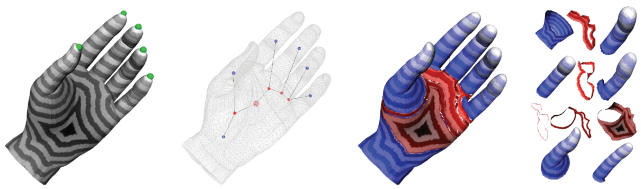


Figure 2: *Segmentation of a hand surface model into its Reeb charts.*

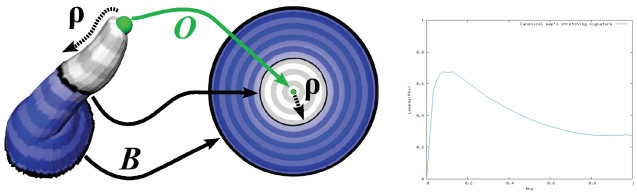


Figure 3: *Disk-like chart unfolding signature computation.*

zero. Thus Reeb charts have the topology of an open annulus (by definition, critical points and, particularly, saddle points equivalence classes are not included in Reeb charts).

Disk-like Reeb charts constitute a specific case. We call a disk-like Reeb chart a Reeb chart which is adjacent to only one local extremum of f. As the related boundary component collapses to a point (the extremum), that kind of chart is given the topology of an open disk.

In Figure 2(d), disk-like Reeb charts have been coloured in blue and annulus-like ones in red. Notice that this decomposition brings a certain *visual semantic*: each of the fingers of the hand model forms a distinct chart.

3.2. Reeb chart unfolding signatures

To achieve efficient pose-insensitive geometrical signature computation for each Reeb chart C_i, we propose to characterize C_i by parameterisation techniques [WWJ*06, TVD07], especially by its mapping ϕ_i to the canonical planar domain D. In particular, we propose to compute C_i unfolding signature [TVD07], a concise vector that describes the evolution of the area distortion introduced by a planar mapping, as summarized below. Thanks to the Reeb graph properties, only two cases had to be considered: disk-like charts and annulus-like charts (cf. statement 1).

Given a disk-like chart C_i, let O be the local extremum of f it contains and B its boundary. We let ϕ_i map O to the centre of the unit planar disk, B to its boundary and f level lines to concentric circles, as shown in Figure 3, where the thumb of the hand of Figure 2 has been mapped to the planar domain D.

Let $\rho(p) \in]0, 1[$ be the normalized absolute difference of f values between O and a point $p \in C_i$, as shown in Figure 3. Consequently to the Reeb chart definition, the sub-level set of ρ ($\{(p_1, p_2 \dots p_n) | \rho(p_1, p_2, \dots p_n) < \rho_0\}$) have also disk topology, as illustrated by the white sub-level set in Figure 3. As ρ increases, the shape of the ρ sub-level sets varies. Thus it induces an evolution in the distortion introduced by their mapping to D. Consequently, to capture the evolution of the sub-level set shape variation, the unfolding signature λ_{ϕ_i} of ϕ_i is defined as follows:

$$\lambda_{\phi_i}(\rho) = \frac{A_{C_i}(\rho)}{A_D(\rho)} = \frac{A_{C_i}(\rho)}{\pi\rho^2}, \tag{1}$$

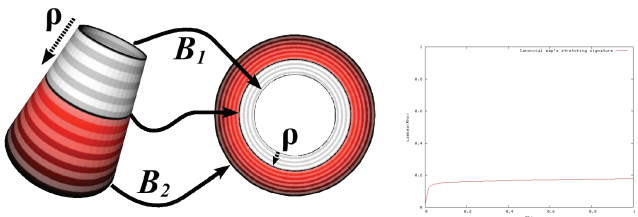
where $A_{C_i}(\rho)$ stands for the area of the sub-level set for parameter ρ on the original surface (C_i) and $A_D(\rho)$ stands for the area of the sub-level set on D. $A_{C_i}(\rho)$ is computed by summing the areas of the related triangles of T (after having normalized edge length by f, similarly to ρ).

Roughly, $\lambda_{\phi_i}(\rho)$ depicts the *stretch* one has to apply on the chart to map it to a disk as ρ increases.

As shown in [TVD07], as f is based on geodesic distances, the parameterisation (and thus its signatures) is invariant to rigid transformations and robust to non-rigid transformations. Moreover, it is also robust to surface noise.

An analog reasoning can be applied for annulus-like charts. Let B_1 be the boundary of shortest perimeter of an annulus-like chart C_j and B_2 the other one. In this case, we let ϕ_j

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.



(a) (b)

Figure 4: *Annulus-like chart unfolding signature.*

map B_1 to the inner boundary of the unit planar annulus and B_2 to its outer boundary, as shown in Figure 4. By defining the ρ parameter similarly to the previous case, the signature λ_{ϕ_j} of ϕ_j is computed as follows (1 is the inner radius of the unit annulus):

$$\lambda_{\phi_j}(\rho) = \frac{A_{C_j}(\rho)}{A_D(\rho)} = \frac{A_{C_j}(\rho)}{\pi(\rho+1)^2 - \pi}. \quad (2)$$

3.3. Implementation details

For feature point extraction, among existing techniques [KLT05, LZ07, TVD08], we used the approach described in [TVD08] as at least two antipodal feature points are always guaranteed to be extracted (so f can always be computed). This technique proposes to identify the extremities of prominent components by intersecting the sets of extrema of two scalar functions based on the geodesic distances to two automatically extracted antipodal vertices [TVD08]. It runs in $O(n \log(n))$ steps with n being the number of vertices in the surface mesh and the extraction is robust to noise and invariant to isometric transformations.

Regarding the Reeb graph computation, the geodesic distance to feature points function does not necessarily fulfill all the requirements of a simple Morse function: (i) smoothness, (ii) distinctly valued critical points and (iii) non-degenerate critical points. Thus, we propose a strategy based on f perturbation to guarantee Reeb graph intrinsic properties (especially about the degree of its nodes). First, (i) f is not differentiable in the configurations of the surface where points are geodesic-equidistant from several feature points. In the discrete setting, this phenomenon leads to the apparition of additional maxima and saddles in these configurations. To cancel these specific critical points, we propose a strategy based on global perturbation, used in [TVD08] on a similar function in the context of skeleton extraction. Starting from the feature points (f minima in this paper), the algorithm iteratively visits the surface mesh by propagation, using a variant of Dijkstra's algorithm (which uses f as weight). Let v be the vertex visited at a given step of such a sweep, the set of vertices candidate for visit (plus the edges linking them) form an upper-value approximation of $f^{-1}(f(v))$ (level line approximation). Then topological variations are encoded in the graph at iterations where connected components of the level line approximation (*contour* approximations) split or merge (see [TVD08] for further details). As the algorithm visits one vertex per iteration, this process is equivalent to considering for each vertex its visit iteration number as f value. This perturbation simulates value uniqueness (ii). Moreover, as additional critical points located in areas where f is non-differentiable are visited iteratively, they are attributed distinctly increasing values, cancelling pairs of extra maxima and saddles (i). Additionally, if a contour approximation splits in $k+1$ sets from an iteration to another with $k \geq 2$ (or if $k+1$ contour approximations merge into one), a multiple k-saddle unfolding strategy [EHZ01] is used to simulate k simple saddles. This technique transforms degenerate critical points into non-degenerate ones (iii) [EHZ01, CMEH*03].

At this stage of the framework, the input closed 2-manifold M is concisely represented by a dual Reeb graph $\mathcal{R}$, whose nodes are accompanied with the unfolding signatures of corresponding Reeb charts. In particular, each unfolding signature is stored as a vector of real values (using a pre-defined number of samples, 64 in all of our experiments).

4. Reeb Pattern-based matching algorithm

The next step of the framework consists in shape partial similarity estimation. Given two input 2-manifolds M_1 and M_2, the goal is to identify their similar sub-parts and to evaluate their geometrical similarity.

4.1. Problem statement

Given two dual Reeb graphs $\mathcal{R}_1$ and $\mathcal{R}_2$, the goal would be to find an optimal injective mapping m^* between a sub-graph $\mathcal{G}_1^* \in \mathcal{R}_1$ and a sub-graph $\mathcal{G}_2^* \in \mathcal{R}_2$ which maximizes a similarity function $S(m^*)$, computed relatively to the geometrical signatures of the nodes matched by m^*.

A brute-force approach to this problem consists in computing the set $\mathcal{M}$ of all the possible injective mappings $m : \mathcal{R}_1 \rightarrow \mathcal{R}_2$, then identifying through an expansion process the common sub-graphs $\mathcal{G}_1^m \in \mathcal{R}_1$ and $\mathcal{G}_2^m \in \mathcal{R}_2$ (such that $m : \mathcal{G}_1^m \rightarrow \mathcal{G}_2^m$ is an isomorphism) and finally evaluating their geometrical similarity $S(m)$. The solution of the problem would be the mapping $m^* \in \mathcal{M}$ that would maximize S. Let n and k be, respectively, the number of nodes of the largest and the smallest of the two dual Reeb graphs to compare. With such an approach, $|\mathcal{M}| = \frac{n!}{(n-k)!}$, which results in an exponential time complexity evaluation process. Moreover, in practice, as underlined by Messmer and Bunke [MB98], graphs representing real world objects may be affected by noise or distortion, motivating the use of error-tolerant matching algorithm, whose complexity is generally even greater than exact algorithms [MB98].

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

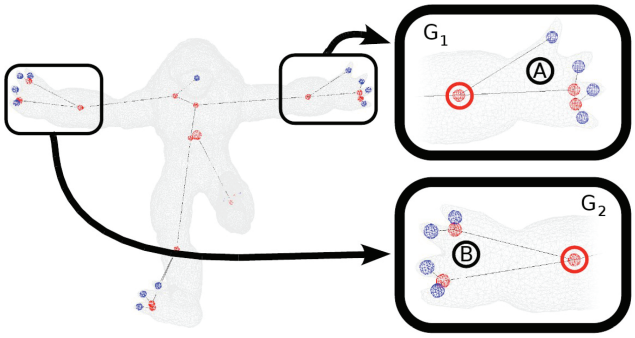


Figure 5: *Structural distortion on two visually similar and topology equivalent Reeb patterns.*

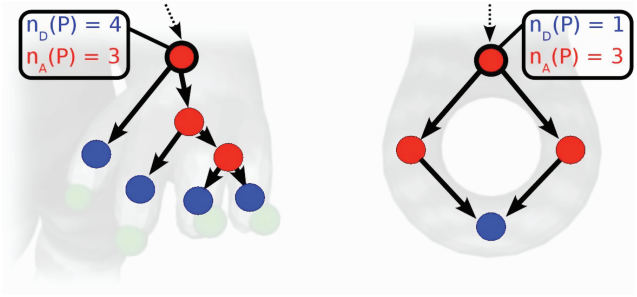


Figure 6: *Structural signatures of a genus-0 and a genus-1 Reeb pattern.*

In the following paragraphs, we describe Reeb graph structural distortion and propose a structural signature for Reeb patterns insensitive to such a distortion (avoiding the need for error-tolerant matching algorithm). Finally, for partial similarity estimation, we dramatically reduce the search space of the problem by considering the set of all the possible injective mappings between topology equivalent Reeb patterns only.

4.2. Reeb pattern and structural distortion

Definition 3 (Reeb pattern) Let M be a compact closed 2-manifold embedded in $\mathbb{R}^3$, $R(f)$ its Reeb graph and C_i an annulus-like Reeb chart. Let B^+ be the boundary component of C_i with highest f value, noted f_{B^+}. Let M^- be the sub-level set of M associated to the f_{B^+} value ($M^- = \{p \in M | f(p) < f_{B^+}\}$). The *Reeb pattern* P_i associated to the annulus-like Reeb chart C_i is the connected component of M^- having B^+ as only boundary component.

Figure 5 shows an Armadillo with its dual Reeb graph and zooms in two Reeb patterns P_1 and P_2 and their related sub-graphs G_1 and G_2. Moreover, the annulus-like charts associated to P_1 and P_2 have been marked with a red circle. Roughly speaking, a Reeb pattern P_i is a surface sub-part which includes nearby protrusions (this is a set of annulus and disk-like charts, delimited by C_i boundary). Notice Reeb patterns are not necessarily defined for each annulus-like Reeb chart (they are not defined for pairs of charts forming handles).

Dual Reeb graphs can suffer from noise and distortion. From our experiences, feature point extraction is very stable on similar objects. This means very few noise due to inexact feature point extraction will appear. However, dual Reeb graphs still suffer from distortion.

Morse theory [Mil63] states that any smooth function on a manifold can be transformed into a Morse function by a slight perturbation, which transforms degenerate critical points into non-degenerate ones. This result depicts the fact that small perturbations on the function can drastically change the properties of the critical points. In particular, in our case, a slight perturbation on the surface (and thus on the function) can change the sequence of bifurcations and junctions in the corresponding dual Reeb graph. As a consequence, in Figure 5, the sub-graph G_1 first bifurcates in the A-labelled region while the sub-graph G_2 first bifurcates in the B-labelled region. Consequently, no isomorphism exists between G_1 and G_2 while the two related Reeb patterns (the hands of the Armadillo) are visually similar and topology equivalent. We refer to this phenomenon as Reeb graph *structural distortion*.

To overcome this issue, in order to compare topology equivalent Reeb patterns despite structural distortion, we introduce the notion of *structural signature* of a Reeb pattern.

Definition 4 (Reeb pattern structural signature) Let M be a compact closed 2-manifold embedded in $\mathbb{R}^3$ and P_i a Reeb pattern associated to an annulus-like Reeb chart C_i. Let $n_D(P_i)$ and $n_A(P_i)$ be, respectively, the number of disk-like and annulus-like Reeb charts included in P_i. The couple $(n_D(P_i), n_A(P_i))$ is the structural signature of P_i.

Statement 2 (Structural signature topological properties) The structural signature of P_i describes P_i topology since $n_D(P_i)$ and $n_A(P_i)$ are linked by the following relation:

$$n_D(P_i) = n_A(P_i) + 1 - 3g_{P_i} \quad (3)$$

with g_{P_i} the genus of the Reeb pattern.

A proof of Equation (3) is given in Appendix A. By definition, all Reeb patterns are 2-manifolds with one boundary. In term of topology description, only their genus differs. Thus their topology is fully described through their structural signature. Moreover, this signature does not encode the way the bifurcations and junctions are sequenced in the Reeb pattern. Thus it does not depict structural distortion.

Figure 6 gives two examples of structural signatures of a genus-0 and a genus-1 Reeb pattern. In the rest of the approach, structural signatures help us both overcoming structural distortion and reducing the search space of the problem.

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

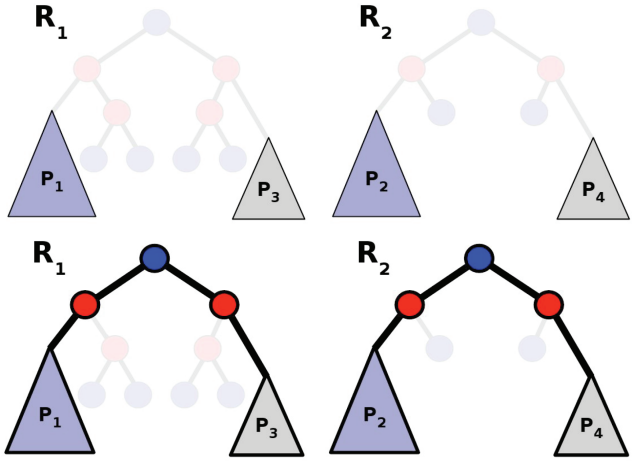


Figure 7: *Expansion example: for a given combination of topology equivalent Reeb patterns ((P_1, P_2) and (P_3, P_4)), the expansion algorithm tries to expand as much as possible the common sub-graphs (in bold in the second row).*

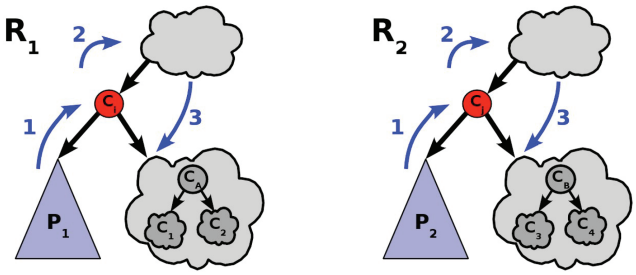


Figure 8: *Expansion process from two topolgy equivalent Reeb patterns P_1 and P_2. Steps **1**, **2** and **3** are depicted by blue arrows.*

4.3. Maximally similar common sub-graph approximation

In the following paragraphs, we introduce an efficient approximation of the brute-force approach that does not consider all the node-to-node combinations but the topology equivalent Reeb pattern to topology equivalent Reeb pattern combinations only.

4.3.1. *Reeb pattern combination enumeration*

First, the structural signature of each Reeb pattern P_i of each of the two dual Reeb graph $\mathcal{R}_1$ and $\mathcal{R}_2$ is computed.

Then, for each identified Reeb pattern P_i, we compute the set of topology equivalent Reeb patterns (whose structural signatures are identical) in the other graph. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be respectively the sets of Reeb patterns of $\mathcal{R}_1$ and $\mathcal{R}_2$ that have a *homologue* in the other graph (whose structural signatures are identical).

Next, we compute the set $\mathcal{M}$ of all the possible injective applications $m : \mathcal{P}_1 \rightarrow \mathcal{P}_2$ that map a Reeb pattern $P_1 \in \mathcal{P}_1$ to a homologue Reeb pattern $P_2 \in \mathcal{P}_2$. In our experiments, the average cardinality of $\mathcal{M}$ is 20 (which has no comparison with the $\frac{n!}{(n-k)!}$ combinations of the brute-force approach).

4.3.2. *Expansion of common sub-graphs*

Next, the common sub-graphs $\mathcal{G}_1^m \in \mathcal{R}_1$ and $\mathcal{G}_2^m \in \mathcal{R}_2$ are built for each m thanks to an expansion algorithm that simultaneously visits $\mathcal{R}_1$ and $\mathcal{R}_2$, starting by Reeb pattern associations. As illustrated in the example of Figure 7, this algorithm expands the initial mapping m by recursively visiting the nodes adjacent to associated Reeb patterns.

Step 1 Considering $\mathcal{R}_1$ and $\mathcal{R}_2$ as directed acyclic graphs (the direction goes towards decreasing f values), the algorithm first visits the *parent nodes* C_i and C_j of P_1 and P_2, as shown in Figure 8 (step **1**). If the degree or the type of C_i and C_j differs, the expansion stops at this point and restarts from another unvisited pair of topology equivalent Reeb patterns. Otherwise, the expansion continues. In the case of orientable 2-manifolds, the degree of an equivalence class of a saddle point in the Reeb graph always equals 3 [Ree46]. As a consequence, two more steps, at most, have to be considered (labelled **2** and **3** in Figure 8).

Step 2 The step **2** recursively repeats steps **1**, **2** and **3** in this order. This means the expansion continues the same way as far as visited nodes have the same degree. Moreover, it stops when nodes corresponding to Reeb patterns explicitly matched by m are reached.

Step 3 In step **3**, the expansion continues with the same stopping condition. However, let C_A and C_B be two nodes that have been matched in step **3**. If their degree equals 3, a decision has to be taken as for the matching of their *children* C_1, C_2 and C_3, C_4 (see Figure 8). *Children* whose Reeb patterns are explicitly matched by m are by definition associated. Remaining nodes are matched according to their degree and their structural signature. At this point, if no decision can be taken according to this criterion, the expansion stops and restarts from another unvisited pair of homologue Reeb patterns.

The output of this algorithm are two common sub-graphs $\mathcal{G}_1^m$ and $\mathcal{G}_2^m$, along with their node and Reeb pattern matches.

4.3.3. *Intra-Reeb pattern node-to-node matching*

The previous algorithm outputs node and Reeb pattern associations. In order to have a full node-to-node correspondence between the two sub-graphs to compare, we have to find a

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

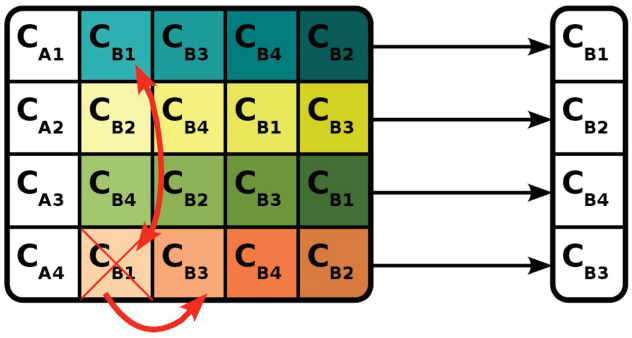


Figure 9: *Intra-Reeb pattern node-to-node matching (bipartite matching approximation).*

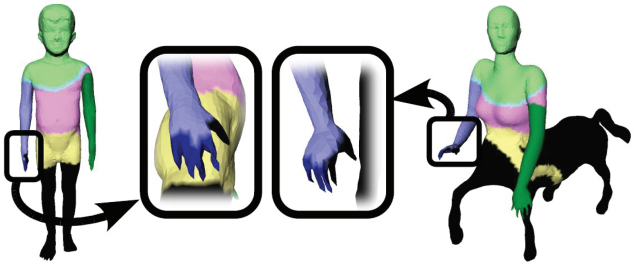


Figure 10: *Reeb chart (bright colors) and pattern (dark colors) matching between a boy and a centaur. Unmatched charts are black. Even though the hands do not have the same number of fingers (some are stuck together on the left model and count for one), they still have been matched.*

node-to-node mapping for each pair of topology equivalent Reeb patterns matched by m (like P_1 and P_2 in Figure 8). Due to possible structural distortion, there might be no isomorphism between the sub-graphs G_1 and G_2 of P_1 and P_2.

Consequently, intra-Reeb pattern node matching is driven by geometrical similarity only running a bipartite matching algorithm, which tries to match the most similar nodes while maximizing the sum of similarities on the patterns. In practice, we run a polynomial approximation of this algorithm as shown in Figure 9. Let $\mathcal{C}_A$ and $\mathcal{C}_B$ be, respectively, the set of disk charts of P_1 and P_2. For each chart $C_{A_i} \in \mathcal{C}_A$, the charts of $\mathcal{C}_B$ are sorted by decreasing values of the geometrical similarity function $s \in [0, 1]$, as shown in the example of Figure 9:

$$s(C_{A_i}, C_{B_j}) = 1 - L_{N1}(C_{A_i}, C_{B_j}), \tag{4}$$

where L_{N1} is the normalized L_1 distance between the unfolding signatures of C_{A_i} and C_{B_j}. Then, the pair of most similar charts are matched. If conflicts occur (see C_{B_1} in the example of Figure 9), only the best matches ((C_{A_1}, C_{B_1}) in the example) are kept unchanged until no more conflict persists. By definition of the Reeb pattern structural signature $|\mathcal{C}_A| = |\mathcal{C}_B|$, thus this algorithm is guaranteed to converge to a solution. Finally, the same process is achieved for the annulus-like charts of P_1 and P_2. At this stage of the algorithm, m is a full node-to-node correspondence between $\mathcal{G}_1^m$ and $\mathcal{G}_2^m$.

4.3.4. *Similarity estimation*

In this step of the comparison, each mapping m is scored relatively to the geometrical similarity S of associated common sub-graphs $\mathcal{G}_1^m$ and $\mathcal{G}_2^m$. At this point, several similarity functions can be defined. In our experiments, we use the following function:

$$S(m) = \frac{\sum_{\forall C_i \in \mathcal{G}_1^m} (1 - L_{N1}(C_i, m(C_i)))^\alpha}{|\mathcal{R}_q|}. \tag{5}$$

In Equation (5), $L_{N1}(C_i, m(C_i))$ stands for the normalized L_1 distance between the unfolding signatures of C_i and $m(C_i)$. α stands for a corrective parameter whose aim is to amplify the geometrical similarity contribution (after training, α is set to 4). Finally $|\mathcal{R}_q|$ stands for the number of nodes of the dual Reeb graph that represents the query model presented to the system. Notice this similarity measure does not respect symmetry or triangular inequality (an object can share similar parts with two other objects that are not similar).

Finally, the mapping $\widehat{m^*}$ that maximizes S is the approximation of the optimal solution m^*. Moreover, $\widehat{\mathcal{G}_1^*}$ and $\widehat{\mathcal{G}_2^*}$ are the approximations of the most similar sub-parts of two 2-manifolds M_1 and M_2, whose partial similarity is given by $S(\widehat{m^*})$. Figure 10 shows an example of the approximated optimal mapping $\widehat{m^*}$ between two sub-graphs $\widehat{\mathcal{G}_1^*}$ and $\widehat{\mathcal{G}_2^*}$ of two surface models, along with the node-to-node and pattern-to-pattern matching.

5. Experiments and Results

To assess the efficiency of the framework, we evaluate its performance on the SHape Retrieval Contest (SHREC) 2007 partial retrieval benchmark [MPB07], and make some comparisons with the methods competing to this contest.

5.1. Partial retrieval benchmark description

This benchmark is composed of a data-set of 400 manifold models (grouped in 20 classes, see Figure 11) and of a query-set of 30 manifold models (see Figure 12). The data-set exhibits diverse variations, from pose change, to shape variability within a same class or topology variation (notice 4 of the 20 classes contain non-zero genus surfaces). Each query shares visually similar sub-parts with at least two classes of the data-set. Figures 13, 14 and 15 show three examples of typical queries and the models of the data-set retrieved by our system. Moreover, for a given query, the ground-truth divides the data-set into *Highly Relevant*, *Marginally Relevant*

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

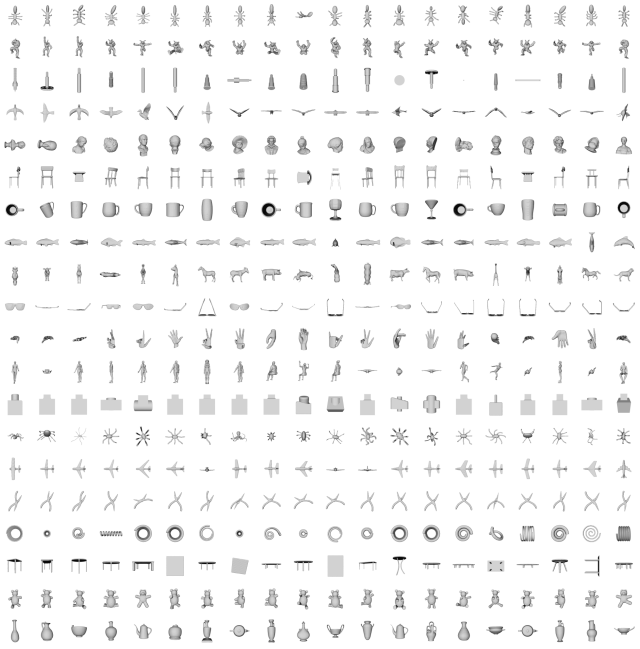


Figure 11: *SHREC 2007 data-set snapshot (one class per row: ants, armadillos, bearings, birds, bustes, chairs, cups, fishes, four-legs, glasses, hands, humans, mechanics, octopuses, planes, pliers, springs, tables, teddies and vases).*

Figure 12: *SHREC 2007 query-set snapshot.*

and *Non-Relevant* class groups. This granularity enables to precisely evaluate the relevance of the results returned by the system. In particular, each relevance group is associated to a specific score, used in the computation of the Normalized Discounted Cumulated Gain (NDCG) vector. Roughly speaking, the higher is the NDCG[i] value, the more relevant are the top-i results. Such a performance measure provides a relevance overview over the whole the data-set. Moreover, it can take into account several classes for scoring a query, which is important for partial similarity since a query can be partially similar to several classes of objects. Furthermore, it is the only performance measurement taken into account in the contest (for more details, we defer the reader to the contest proceedings [MPB07]).

5.2. Experimental setup

The 400 models of the data-set are first indexed off-line. During the off-line process, the dual Reeb graph and the related unfolding signatures (using 64 samples) are computed for each model and stored into the index file of the data-set.

During the on-line process, the dual Reeb graph and the unfolding signatures of the query are first extracted. Then, its dual Reeb graph is compared with each graph of the index as described previously. Finally, the entries of the data-set are sorted by decreasing values of partial similarity. Depending on the number of vertices in the query surface mesh and the size of the dual Reeb graph, the full processing time of a query varies from *4 to 30 seconds* on a 3GHz P4 PC.

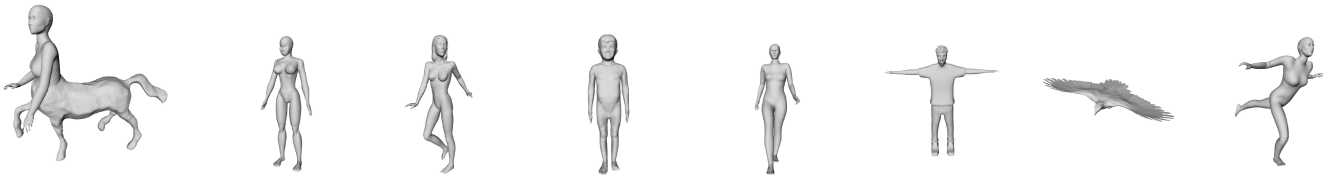
5.3. Framework performance evaluation

First, from a qualitative point of view, Figures 13, 14 and 15 give a good overview of the efficiency of the framework. For example, in Figure 13, the query is a centaur (half-horse, half-human) and thus most of the top-results are humanoid models (first horses are retrieved at rank 15), even if they are not globally similar to the query. Figure 14 demonstrates the method's ability to handle non-zero genus surfaces (the case of the two outliers, rank 6 and 7, is discussed in the limitation dedicated Section 5.5).

From a more quantitative point of view, in the first experiment, we compare the average NDCG vector of our approach with those of the methods competing the contest [CDS*05, BMSF06] (as reported in the contest proceedings [MPB07]). Such a vector is the average of the 30 NDCG vectors corresponding to the 30 models of the query-set. Figure 16 shows the curves corresponding to these vectors. As the RPU curve is higher than the others, it is obvious that it outperforms related methods. Moreover, to quantify its improvement, we introduce the NDCG vector gain G, which is the ratio of the area between two curves and the area below the lowest one:

$$G(A, B) = \frac{\sum_{i=1}^{400}(NDCG[i]_A - NDCG[i]_B)}{\sum_{i=1}^{400} NDCG[i]_B}. \quad (6)$$

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

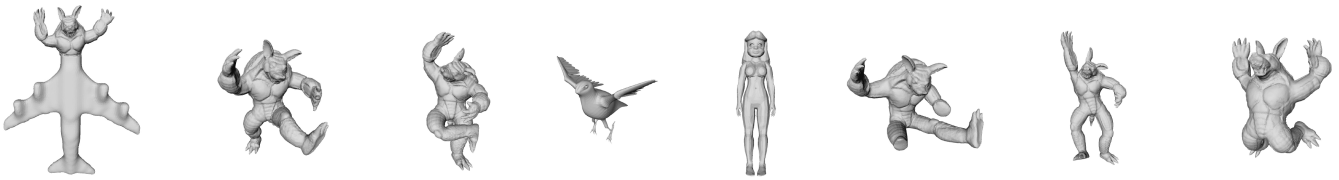


(a) **Query** (b) 1. S = 0.284 (c) 2. S = 0.276 (d) 3. S = 0.247 (e) 4. S = 0.239 (f) 5. S = 0.227 (g) 6. S = 0.226 (h) 7. S = 0.225

Figure 13: *A query from the SHREC 2007 query-set (a centaur) and the top-7 results retrieved by the system.*

(a) **Query** (b) 1. S = 0.265 (c) 2. S = 0.235 (d) 3. S = 0.231 (e) 4. S = 0.211 (f) 5. S = 0.198 (g) 6. S = 0.178 (h) 7. S = 0.159

Figure 14: *A genus-1 query from the SHREC 2007 query-set and the top-7 results retrieved by the system.*



(a) **Query** (b) 1. S = 0.296 (c) 2. S = 0.257 (d) 3. S = 0.256 (e) 4. S = 0.235 (f) 5. S = 0.199 (g) 6. S = 0.197 (h) 7. S = 0.196

Figure 15: *Another query from the SHREC 2007 query-set (an Armadillo composed with a plane) and the top-7 results retrieved by the system. Notice that Armadillos have been retrieved despite their different poses.*

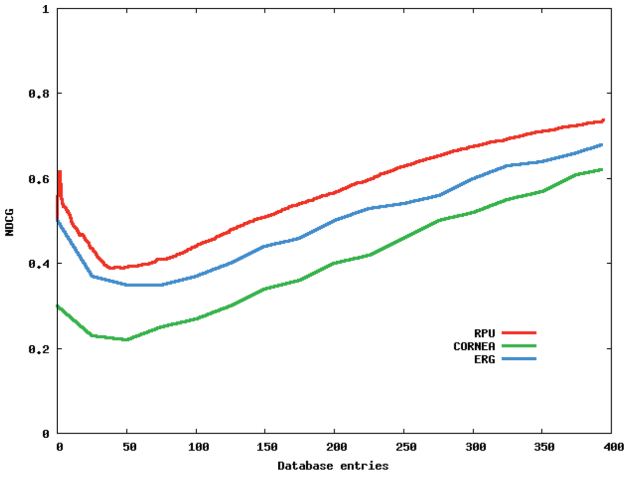


Figure 16: *Average normalized discounted cumulated gain (NDCG) vectors for Reeb pattern unfolding (RPU), Extended Reeb Graphs (ERG) and curve-skeleton based many-to-many matching (CORNEA) on the SHREC 2007 data-set.*

With such a measure, the gain on the methods by Biasotti *et al.* [BMSF06] and Cornea *et al.* [CDS*05] is, respectively, of *14.1%* and *40.9%*. In the second experiment, we evaluate the impact of the choice of the similarity function S (see Section 4.3.4) on the retrieval performances. In particular, we consider the function S_1 that returns the size of the common sub-graphs (the maximally similar common sub-graphs become then the maximum common sub-graphs). We also consider the function S_2 where the unfolding signatures have been replaced by the geometrical attributes used by Hilaga *et al.* [HSKK01]. The related curves are reported in Figure 17. This figure shows that performances comparable to the other methods are obtained with S_2. Moreover, it clearly demonstrates the gain provided by unfolding signatures.

5.4. Robustness evaluation

As the dual Reeb graph construction and the unfolding signature computation are based on normalized geodesic distance evaluation, the approach is guaranteed to be invariant against rigid-transformations. Moreover, Figure 15 demonstrates its robustness against non-rigid transformations, since Armadillos in different poses have been retrieved as top results. In the third experiment, we investigate the framework robustness against surface noise. In particular, for each element of the query-set, we added a surface noise whose amplitude is bounded by, respectively, $\pm 0.5\%$ and $\pm 1\%$ of the lengths of the bounding box of the model, as shown in Figure 18. The

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

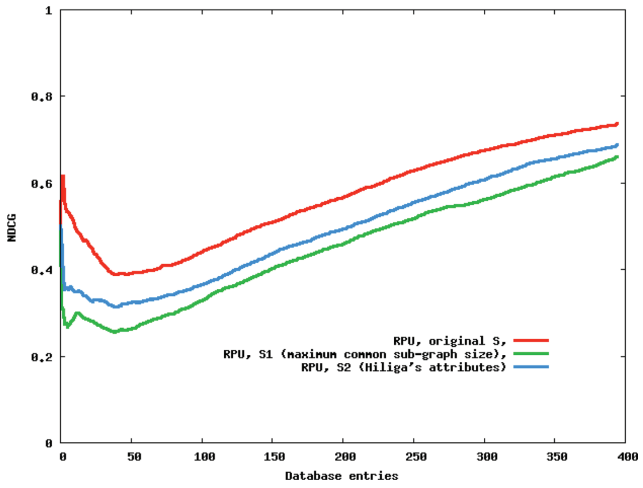


Figure 17: *Contribution of the chart unfolding signatures to the performances.*

(a) Original model (b) Noise $\pm 0.5\%$ (c) Noise $\pm 1\%$

Figure 18: *Surface noise on a SHREC 2007 query.*

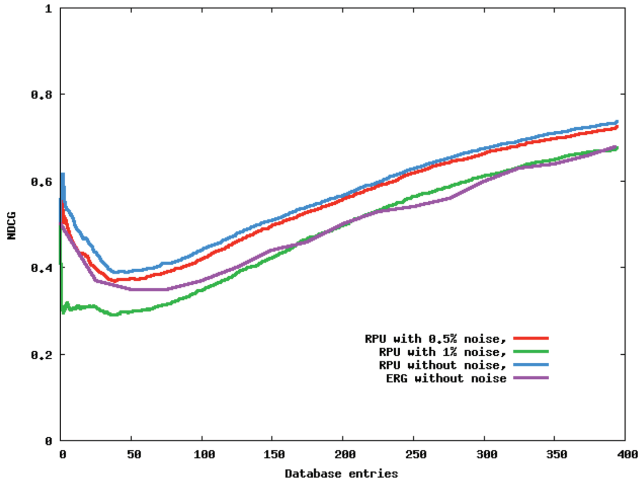


Figure 19: *Robustness evaluation of RPU with a noisy version of the SHREC 2007 query-set. Even with a surface noise of* $\pm$ 0.5%, *RPU still outperforms state-of-the-art techniques scores on clean data.*

NDCG vectors with such corrupted query-sets have been reported in Figure 19. These curves demonstrate the stability of the algorithm despite the noise. Moreover, even with a surface noise of $\pm$ 0.5%, it stills outperforms ERG [BMSF06] scores on clean data.

5.5. Discussion and limitations

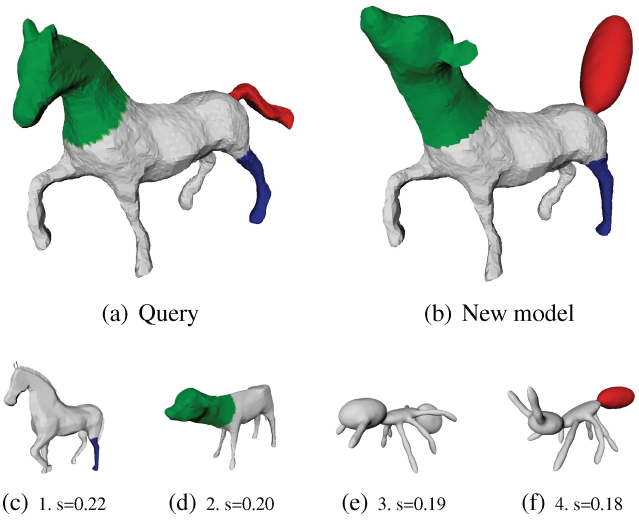
As other structural based approaches, the surface decomposition step of the framework introduces a bias in the comparison process. To guarantee stability and performance, this decomposition has to be stable within a same class of objects and moreover coherent with the data-set ground-truth. The bias introduced in the presented technique is based on feature point extraction (which drives the surface segmentation). Hence, this stage is a critical part of the framework and impacts its performances. In practice, with the SHREC data-set, feature extraction turns out to be homogeneous within most classes. As a counter-example, the *four legs* class is composed of models representing distinct types of animals, having different protrusions (with or without tail, horns, etc.) and thus leads to slightly distinct decompositions. In the future, we would like to investigate other decomposition strategies that would overcome this issue but still take into account local shape features.

Moreover, as the decomposition and matching processes are topology based, the overall framework is quite sensible to topology variations within a same class of objects. For example, most of humanoids are modelled with genus zero surfaces but if a humanoid's feet are stuck to each other the related surface will have genus one and consequently the legs will mainly match with objects having handles. Same remarks go to the example of Figure 14, where the topology influences the results: the genus-1 part of the query is matched with the back of the chair (rank 6) while its left bent appendix is matched with one of the chair's (rank 6) and table's (rank 7) legs (the unfolding signatures are bending insensitive). In the future, the combination of local and structural approaches is worth being studied to overcome this issue.

6. Application to Modelling by Example

To demonstrate the applicative interest of the framework, we designed an intelligent modelling-by-example system [FKS*04]. Figures 20 and 21 show typical use examples. First, the user queries the data-set through our partial shape retrieval system. Then, he/she selects on the query the Reeb patterns (or Reeb pattern sub-parts) he/she wants to remove (in red, green and blue in figure 20(a)). Next, the system automatically highlights the parts of the retrieved results candidates for swapping, based on our graph matching algorithm. The user then selects the retrieved results and cut the desired parts (highlighted or not, figures 20(c), 20(d) and 20(f)). After scaling, rotating and translating the parts according to his/her needs, the cut sub-parts are pasted back onto the query model (figure 20(b)). In our experiments, boundaries are simply re-meshed but more visually appealing methods can be employed for gluing parts [SBSCO06]. Thanks to the partial shape retrieval and the part swapping suggestions, a novice user can intuitively and rapidly create new 3D shapes

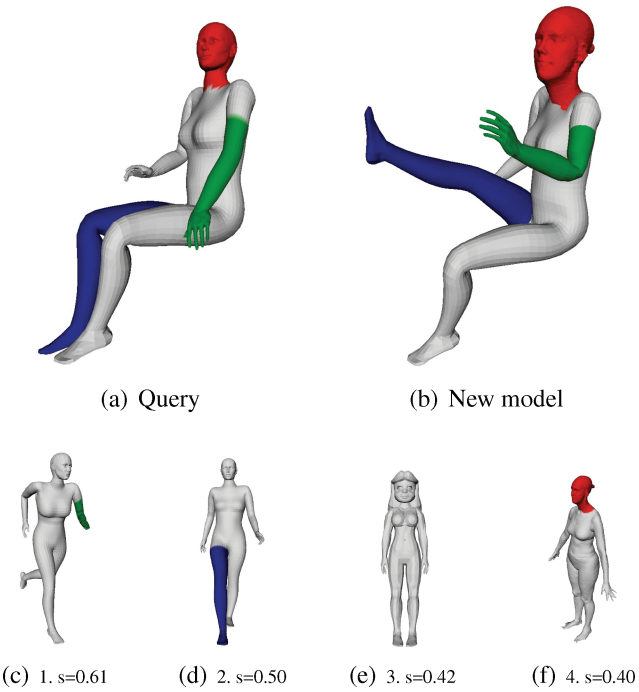
© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.



(a) Query (b) New model

(c) 1. s=0.22 (d) 2. s=0.20 (e) 3. s=0.19 (f) 4. s=0.18

Figure 20: *Modelling a cow-horse by example: the user selects on the query, then on the retrieved results (second row), the Reeb patterns to be exchanged.*



(a) Query (b) New model

(c) 1. s=0.61 (d) 2. s=0.50 (e) 3. s=0.42 (f) 4. s=0.40

Figure 21: *Modelling a new woman model by example, composing both synthetic and scanned models.*

with high geometric details based on the example parts of interest retrieved by our system.

7. Conclusion

In this paper, we presented a novel approach for fast and efficient partial 3D shape retrieval. The main contribution of this work was to take advantage of Reeb graph theory properties to improve both the shape description and comparison processes. Extensive experiments demonstrated the improvement of each of these steps, resulting in an overall gain of, respectively, *14.1%* and *40.9%* on the methods by Biasotti *et al.* [BMSF06] and Cornea *et al.* [CDS*05]. Moreover, the robustness to rigid and non-rigid transformations and surface noise has been shown. Finally, queries are processed in interactive time (from 4 to 30 seconds).

However, shape topological description might be too discriminant in some particular cases. For example, humanoids can be modelled with closed fists or open hands, resulting in different Reeb graphs and thus penalized partial similarity. This is a major drawback of topology-based approaches. In the future, the combination of local and structural approaches is worth being studied.

Moreover, input surface models are required to be manifold. In the future, we would like to adapt the framework to more general mesh models. Reeb graph construction algorithms for non-manifold surfaces can be a starting point [PSBM07].

Finally, as dynamic meshes (3D plus time) are becoming more and more popular, in the future, we would like to investigate the use of Reeb graphs of higher dimension manifolds [EHMP04] for the partial retrieval of 3D dynamic shapes. We believe this research topic is one of the next challenges of the shape retrieval community and could result in nice user applications like *animation-by-example* systems.

Acknowledgements

The SHREC data-set is courtesy of AIM@SHAPE. A big extra special super thank goes to Simone Marini (IMATI) for his help on the benchmark. We also thank the anonymous reviewers for their help on the paper, especially for their inspiring remarks about the proof of appendix A. This work is partially supported by the ANR (Agence Nationale de la Recherche) through MADRAS project (ANR-07-MDCO-015).

Appendix A: $n_D(P) = n_A(P) + 1 - 3g_P$

Proof Let f be a simple Morse function defined on a compact closed and orientable 2-manifold M and $R(f)$ its Reeb graph.

Let P be a Reeb pattern defined relatively to $R(f)$. By definition, P is a compact 2-manifold with one boundary component.

Let P' be the compact and orientable 2-manifold without boundary obtained by the closure of the boundary of P (where the unique boundary component of P is glued by contraction to a point, as shown in Figures 22 and 23). Let f' be a simple Morse function defined on P' such that it has the

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

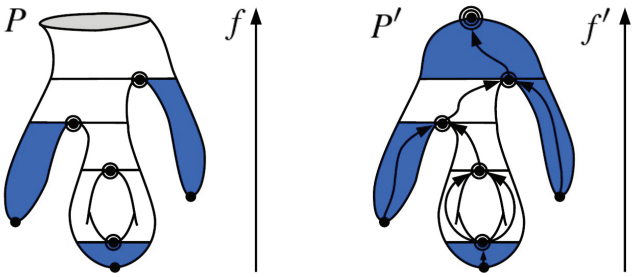


Figure 22: *Enumeration of the disk-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P. Right: Reeb pattern P′ after the closure of the unique boundary component of P. Disk-like Reeb charts are in blue.*

Annulus-like Reeb chart enumeration

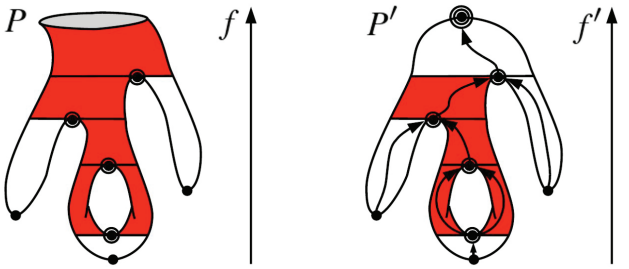


Figure 23: *Enumeration of the annulus-like Reeb charts of a Reeb pattern. Left: Original Reeb pattern P. Right: Reeb pattern P′ after the closure of the unique boundary component of P. Annulus-like Reeb charts are in red.*

same set of critical points and critical values on P' than f on P, plus one maximum (due to the closure of the boundary component of P). f' has distinctly valued critical points and all its critical points are non-degenerate. f' is indeed a simple Morse function. For example, in Figures 22 and 23, f and f' are the height functions. Let $R(f')$ be the Reeb graph of f' (depicted by black arrows on Figures 22 and 23, right).

On the first hand, the Euler characteristic $\chi(P')$ of P' is given by the generalized Euler relation [FK97] (where $g_{P'}$ is the genus of P'):

$$\chi(P') = 2 - 2g_{P'}. \tag{7}$$

On the other hand, the Euler characteristic $\chi(P')$ of P' can also be expressed by the Morse-Euler formula [FK97] since P' is a closed manifold (where $k = 2$ is the dimension of the manifold):

$$\chi(P') = \sum_{i=0}^{k} (-1)^i \mu_i(f') = \mu_0(f') - \mu_1(f') + \mu_2(f'). \tag{8}$$

In Equation (8), $\mu_i(f')$ stands for the i^{th} Morse number of f', which is equal to the number of f' critical points of index i. In particular, $\mu_0(f')$, $\mu_1(f')$ and $\mu_2(f')$ are, respectively, the number of f' local minima, saddles and maxima. In Figures 22 and 23, each critical point is marked with a number of concentric circles equal to its index i.

Disk-like Reeb chart enumeration

Let $n_D(P')$ be the number of disk-like Reeb charts of P'. By definition, each disk-like Reeb chart of P' is exactly adjacent to one local extremum of f'. Moreover, the degree in $R(f')$ of equivalence classes of minima and maxima equals one [Ree46] (see Figure 22, right). Then, a local extremum of f' can only be adjacent to one disk-like Reeb chart of P'. Thus

$$n_D(P') = \mu_0(f') + \mu_2(f'). \tag{9}$$

Moreover, by definition, f' has exactly the same set of critical points and critical values on P' than f on P, plus one maximum due to the closure of the boundary component of P (see Figure 22). Consequently, P and P' have the same Reeb chart decomposition at the exception of the chart adjacent to the additional maximum of f', which is transformed from an annulus-like Reeb chart (in P, Figure 22, left) to a disk-like Reeb chart (in P', Figure 22, right) by the closure of the unique boundary component of P. Hence

$$n_D(P) = n_D(P') - 1 = \mu_0(f') + \mu_2(f') - 1. \tag{10}$$

Annulus-like Reeb chart enumeration

Let $n_A(P')$ be the number of annulus-like Reeb charts of P'. $R(f')$ is a finite and connected 1-D simplicial complex [Ree46]. Consequently, it can be considered as a planar graph, whose vertices are the equivalence classes corresponding to critical points of f' and whose edges correspond to the Reeb charts of P'. Then, the Euler relation for planar graphs holds

$$\chi(R(f')) = V_{R(f')} - E_{R(f')} + F_{R(f')} = 2, \tag{11}$$

where $V_{R(f')}$ is the number of critical points of f':

$$V_{R(f')} = \mu_0(f') + \mu_1(f') + \mu_2(f'), \tag{12}$$

where $E_{R(f')}$ is the number of Reeb charts of P':

$$E_{R(f')} = n_D(P') + n_A(P') \tag{13}$$

and where $F_{R(f')}$ is the number of faces of the planar graph (with $L_{R(f')}$ the number of loops in $R(f')$):

$$F_{R(f')} = L_{R(f')} + 1. \tag{14}$$

Moreover, $R(f')$ has $g_{P'}$ loops [CMEH*03] (with $g_{P'}$ the genus of P'). Then, Equations (11), (12), (13) and (14) become:

$$\begin{aligned}\chi(R(f')) = 2 = \mu_0(f') + \mu_1(f') + \mu_2(f') \\ - (n_D(P') + n_A(P')) + g_{P'} + 1.\end{aligned} \tag{15}$$

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

Thanks, to Equation (9), we have:

$$2 = \mu_1(f') - n_A(P') + g_{P'} + 1$$
$$n_A(P') = \mu_1(f') + g_{P'} - 1. \quad (16)$$

Moreover, by definition, f' has exactly the same set of critical points and critical values on P' than f on P, plus one maximum due to the closure of the unique boundary component of P (see Figure 23). Consequently, P and P' have the same Reeb chart decomposition at the exception of the chart adjacent to the additional maximum of f', which is transformed from an annulus-like Reeb chart (in P, Figure 23, left) to a disk-like Reeb chart (in P', Figure 23, right) by the closure of the unique boundary component of P. Hence

$$n_A(P) = n_A(P') + 1$$
$$n_A(P) = \mu_1(f') + g_{P'}. \quad (17)$$

Relation between $n_D(P)$ and $n_A(P)$

Consequently to Equations (10) and (17), Equation (8) becomes:

$$\chi(P') = \mu_0(f') - \mu_1(f') + \mu_2(f')$$
$$\chi(P') = n_D(P) + 1 - (n_A(P) - g_{P'}). \quad (18)$$

Therefore, thanks to the Equation (7), we have the following relations:

$$n_D(P) + 1 - n_A(P) + g_{P'} = 2 - 2g_{P'}$$
$$n_D(P) = n_A(P) + 1 - 3g_{P'}. \quad (19)$$

Moreover, by definition, the boundary of a Reeb pattern is required to be composed of a single connected component. Then, the boundary of a Reeb pattern cannot lie on a topological handle. Consequently, the closure of the unique boundary component of P cannot modify its genus g_P. Then

$$g_P = g_{P'}. \quad (20)$$

Thanks to the Equation (20), we have the final result:

$$n_D(P) = n_A(P) + 1 - 3g_P. \quad (21)$$

References

[AHLD07] Aujay G., Hétroy F., Lazarus F., Depraz C.: Harmonic skeletons for realistic character animation. In *Symposium on Computer Animation* (2007), pp. 151–160.

[BGSF08] Biasotti S., Giorgi D., Spagnuolo M., Falcidieno B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392* (2008), 5–22.

[BKS*05] Bustos B., Keim D. A., Saupe D., Schreck T., Vranic D. V.: Feature-based similarity search in 3D object databases. *ACM Computing Surveys 37* (2005), 345–387.

[BMS00] Biasotti S., Mortara M., Spagnuolo M.: Surface compression and reconstruction using Reeb graphs and shape analysis. In *Spring Conference on Computer Graphics* (2000), pp. 175–184.

[BMSF06] Biasotti S., Marini S., Spagnuolo M., Falcidieno B.: Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design Journal 38* (2006), 1002–1019.

[CDS*05] Cornea N. D., Demirci M. F., Silver D., Shokoufandeh A., Dickinson S., Kantor P. B.: 3D object retrieval using many-to-many matching of curve skeletons. In *IEEE Shape Modeling International* (2005), pp. 366–371.

[CMEH*03] Cole-mclaughlin K., Edelsbrunner H., Harer J., Natarajan V., Pascucci V.: Loops in Reeb graphs of 2-manifolds. In *Symposium on Computational Geometry* (2003), pp. 344–350.

[CSM03] Cohen-steiner D., Morvan J.-M.: Restricted delaunay triangulations and normal cycle. In *Symposium on Computational Geometry* (2003), pp. 312–321.

[CTSO03] Chen D. Y., Tian X. P., Shen Y. T., Ouhyoung M.: On visual similarity based 3D model retrieval. *Computer Graphics Forum 22* (2003), 223–232.

[DJ06] Dey T. K., Jian S.: Defining and computing curve skeletons with medial geodesic function. In *Eurographics Symposium on Geometry Processing* (2006), pp. 143–152.

[EHMP04] Edelsbrunner H., Harer J., Mascarenhas A., Pascucci V.: Time-varying Reeb graphs for continuous space-time data. In *Symposium on Computational Geometry* (2004), pp. 366–372.

[EHZ01] Edelsbrunner H., Harer J., Zomorodian A.: Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. In *Symposium on Computational Geometry* (2001), pp. 70–79.

[FK97] Fomenko A., Kunii T.: *Topological Modeling for Visualization*. Springer-Verlag, 1997.

[FKS*04] Funkhouser T., Kazhdan M., Shilane P., Min P., Kiefer W., Tal A., Rusinkiewicz S., Dobkin D.: Modeling by example. *ACM Transactions on Graphics 23* (2004), 652–663.

[FMK*03] Funkhouser T., Min P., Kazhdan M., Chen J., Halderman A., Dobkin D.: A search engine for 3D models. *ACM Transactions on Graphics 22* (2003), 83–105.

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

[FS06] Funkhouser T., Shilane P.: Partial matching of 3D shapes with priority-driven search. In *Eurographics Symposium on Geometry Processing* (2006), pp. 131–142.

[GCO06] Gal R., Ccohen-Or D.: Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics 25* (2006), 130–150.

[GSCO07] Gal R., Shamir A., Cohen-Or D.: Pose oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics 13* (2007), 261–271.

[HKDH04] Huber D., Kapuria A., Donamukkala R., Hebert M.: Parts-based 3D object classification. In *IEEE Computer Vision and Pattern Recognition* (2004), pp. 82–89.

[HSKK01] Hilaga M., Shinagawa Y., Kohmura T., Kunii T.: Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH* (2001), pp. 203–212.

[IJL*05] Iyer N., Jayanu S., Lou K., Kalyanaraman Y., Ramani K.: Three-dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design Journal 37* (2005), 509–530.

[JH99] Johnson A. E., Hebert M.: Using spin-images for efficient multiple model recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21* (1999), 433–449.

[JZ07] Jain V., Zhang H.: A spectral approach to shape-based retrieval of articulated 3D models. *Computer-Aided Design Journal 39* (2007), 398–407.

[KLT05] Katz S., Leifman G., Tal A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 21* (2005), 865–875.

[KSDD03] Keselman Y., Shokoufandeh A., Demirci M., Dickinso S.: Many-to-many graph matching via metric embedding. In *IEEE Computer Vision and Pattern Recognition* (2003), pp. 850–857.

[LZ07] Liu R., Zhang H.: Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum 26* (2007), 385–394.

[LZQ06] Liu Y., Zha H., Qin H.: Shape topics: a compact representation and new algorithms for 3D partial shape retrieval. In *IEEE Computer Vision and Pattern Recognition* (2006), pp. 2025–2032.

[MB98] Messmer B. T., Bunke H.: A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20* (1998), 493–504.

[MGP06] Mitra N. J., Guibas L., Pauly M.: Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics 25* (2006), 560–568.

[Mil63] Milnor J.: *Morse Theory*. Princeton University Press, 1963.

[MPB07] Marini S., Paraboschi L., Biasotti S.: SHape Retrieval Contest 2007: Partial matching track. In *SHREC (in conjunction with IEEE Shape Modeling International)* (2007), pp. 13–16.

[NGH04] Ni X., Garland M., Hart J.: Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics 23* (2004), 613–622.

[PSBM07] Pascucci V., Scorzelli G., Bremer P. T., Mascarenhas A.: Robust on-line computation of Reeb graphs: Simplicity and speed. *ACM Transactions on Graphics 26* (2007), 58.1–58.9.

[Ree46] Reeb G.: Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes-rendus des Séances de l'Académie des Sciences 222* (1946), 847–849.

[SBSCO06] Sharf A., Blumenkrants M., Shamir A., Cohen-Or D.: Snappaste: An interactive technique for easy mesh composition. *The Visual Computer 22* (2006), 835–844.

[TS05] Tung T., Schmitt F.: The augmented multiresolution Reeb graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modeling 11* (2005), 91–120.

[TV04] Tangelder J. W. H., Veltkamp R. C.: A survey of content based 3D shape retrieval methods. In *IEEE Shape Modeling International* (2004), pp. 145–156.

[TVD07] Tierny J., Vandeborre J.-P., Daoudi M.: Reeb chart unfolding based 3D shape signatures. In *Eurographics* (2007), pp. 13–16.

[TVD08] Tierny J., Vandeborre J.-P., Daoudi M.: Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer 24* (2008), 155–172.

[WWJ06] Wang S., Wang Y., Jin M., Gg X., Samaras D.: 3D surface matching and recognition using conformal geometry. In *IEEE Computer Vision and Pattern Recognition* (2006), pp. 2453–2460.

[ZMT05] Zhang E., Mischaikow K., Turk G.: Feature-based surface parametrization and texture mapping. *ACM Transactions on Graphics 24* (2005), 1–27.

© 2008 The Authors
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

Annexe D

A comparative study of existing metrics for 3D-mesh segmentation evaluation. Halim Benhabiles, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. *The Visual Computer - International Journal of Computer Graphics*, 26(12) :1451–1466, December 2010.

Abstract. In this paper, we present an extensive experimental comparison of existing similarity metrics addressing the quality assessment problem of mesh segmentation. We introduce a new metric, named the 3D Normalized Probabilistic Rand Index (3D-NPRI), which outperforms the others in terms of properties and discriminative power. This comparative study includes a subjective experiment with human observers and is based on a corpus of manually segmented models. This corpus is an improved version of our previous one (Benhabiles et al. in IEEE International Conference on Shape Modeling and Application (SMI), 2009). It is composed of a set of 3D-mesh models grouped in different classes associated with several manual ground-truth segmentations. Finally the 3D-NPRI is applied to evaluate six recent segmentation algorithms using our corpus and the Chen et al.'s (ACM Trans. Graph. (SIGGRAPH), 28(3), 2009) corpus.

Vis Comput
DOI 10.1007/s00371-010-0494-2

ORIGINAL ARTICLE

A comparative study of existing metrics for 3D-mesh segmentation evaluation

Halim Benhabiles · Jean-Philippe Vandeborre · Guillaume Lavoué · Mohamed Daoudi

© Springer-Verlag 2010

Abstract In this paper, we present an extensive experimental comparison of existing similarity metrics addressing the quality assessment problem of mesh segmentation. We introduce a new metric, named the 3D Normalized Probabilistic Rand Index (3D-NPRI), which outperforms the others in terms of properties and discriminative power. This comparative study includes a subjective experiment with human observers and is based on a corpus of manually segmented models. This corpus is an improved version of our previous one (Benhabiles et al. in IEEE International Conference on Shape Modeling and Application (SMI), 2009). It is composed of a set of 3D-mesh models grouped in different classes associated with several manual ground-truth segmentations. Finally the 3D-NPRI is applied to evaluate six recent segmentation algorithms using our corpus and the Chen et al.'s (ACM Trans. Graph. (SIGGRAPH), 28(3), 2009) corpus.

Keywords 3D-mesh segmentation · Ground-truth · Similarity metric · Subjective tests · Evaluation

H. Benhabiles (✉) · J.-P. Vandeborre · M. Daoudi
LIFL (UMR USTL/CNRS 8022), University of Lille, Lille, France
e-mail: halim.benhabiles@lifl.fr

J.-P. Vandeborre · M. Daoudi
Institut TELECOM; TELECOM Lille 1, Lille, France

J.-P. Vandeborre
e-mail: jean-philippe.vandeborre@lifl.fr

M. Daoudi
e-mail: mohamed.daoudi@lifl.fr

G. Lavoué
CNRS, INSA-Lyon, LIRIS, UMR5205, Université de Lyon, 69621, Lyon, France
e-mail: glavoue@liris.cnrs.fr

Published online: 17 April 2010

1 Introduction

3D-mesh segmentation is a fundamental process in many applications such as shape retrieval [1, 29], compression [29], deformation [17], texture mapping [26], etc. It consists in decomposing a polygonal surface into different regions (i.e. connected set of vertices or facets) of uniform properties, either from a *geometric* point of view or from a *semantic* point of view. It is a critical step toward content analysis and mesh understanding. Although some supervised methods exist [12, 16], most existing techniques are fully automatic.

According to recent states-of-the-art [3, 24], mesh segmentation techniques can be classified into two categories: surface-type (or *geometric*) methods and part-type (or *semantic*) methods. In the first case, the algorithms are based on low level geometric information (e.g. curvature [20]) in order to define segments (i.e. regions) with respect to geometric homogeneity, while in the latter case, the algorithms aim at distinguishing segments that correspond to relevant features of the shape, by following higher level notions such as defined in human perception theory [6]. This kind of approach is particularly suited for object animation/deformation and indexing applications, where the decomposition has to be meaningful.

Although development of mesh segmentation algorithms for both approaches has drawn extensive and consistent attention, relatively little research has been done on segmentation *evaluation*. For the first approach (surface-type), some tools exist depending on the end application as texture mapping [23] or medical imaging [13]. Recently, two main works, Benhabiles et al. [4] (our previous work) and Chen et al. [7], have been proposed to study the quality assessment problem of part-type 3D-mesh segmentation. Both works propose a benchmark for segmentation evaluation which is

based on a ground-truth corpus. The corpus is composed of a set of 3D-models grouped in different classes and associated with several manual segmentations produced by human observers. These two benchmarks comprise the ground-truth corpus and a set of similarity metrics, then the evaluation of a segmentation algorithm consists in measuring the similarity between the reference segmentations from the corpus and that obtained by this algorithm (on the same models). In this kind of benchmark the quality of the evaluation depends on the quality of the corpus but also on the quality of the segmentation similarity measure. This leads to conclude that the choice of an accurate measure is quite critical in order to provide a strict evaluation and to reflect the real quality of an automatic segmentation with comparison to a manual one. In this context, less efforts were investigated to propose a reliable measure of mesh segmentation similarity. Indeed, the previous works [4, 7] focused their interests on the design of the ground-truth corpus and presented rather simple metrics suffering from degeneracies and low discriminative power.

In this context the objective of the present work is to evaluate the existing metrics and to propose a new one which is more reliable. This paper introduces three main contributions. Firstly, we propose a thorough study and comparisons of existing metrics addressing the assessment problem of mesh segmentation, using a corpus of manually segmented models. This corpus is an improved version of our previous one [4] and is available on-line.[1] Secondly, we propose a new measure of segmentation similarity that allows one to quantify the consistency between multiple segmentations of a model. We show that this new metric outperforms existing ones in terms of properties and discriminative power. To quantitatively compare the discriminative power of the metrics, we have conducted subjective tests using a set of human observers. Thirdly, we apply this measure together with two corpuses (our corpus and Chen et al.'s [7] corpus) to evaluate six recent 3D-mesh segmentation algorithms.

This paper is organized as follows. In Sect. 2, we provide a review of the state-of-the-art of segmentation evaluation and an analytic study of the measures that have been proposed in this context. In Sects. 3 and 4, we define a new objective metric to perform a quantitative comparison between a segmentation algorithm and a set of ground-truth segmentations (of the same model). In Sect. 5, we present our corpus which will be used for the experimental comparison of the metrics and for the evaluation of the segmentation algorithms. In Sect. 6, we present an extensive experimental comparison between our new metric and existing ones, then we analyze the discriminative power of this new metric using subjective tests. In Sect. 7, we demonstrate the usability of our whole evaluation protocol through the evaluation of six recent segmentation methods. Section 8 concludes the paper.

[1] http://www-rech.telecom-lille1.eu/3dsegbenchmark/.

2 Related work

In this section, we firstly provide a review of the state-of-the-art of 2D-image and 3D-mesh segmentation evaluation. Indeed, the most significant works for the 3D-mesh segmentation evaluation [4, 7] are based on the same methodology as that proposed in the 2D-image domain [21]. Secondly, we review the measures that have been proposed in the context of 3D-mesh segmentation evaluation, while analyzing their properties.

2.1 State-of-the-art of 2D-image and 3D-mesh segmentation evaluation

Several advanced works exist for the quality assessment of *2D-image* segmentation. Zhang et al. [30] offer a study on the different methods proposed for this task. According to them, the different methods can be classified into five groups:

- *Analytical methods*. They directly treat the segmentation algorithms themselves by taking into account principles, requirements, utilities, complexity, etc. of algorithms. Using analytical methods to evaluate segmentation algorithm avoids a concrete implementation of the algorithms. However, the real quality of these algorithms cannot be obtained by a simple analytical study.
- *Subjective methods*. They evaluate the segmentation algorithms in a subjective way in which the segmentation results are judged by a human operator. Therefore, the evaluation scores can vary significantly from one human evaluator to another since they do not have necessarily the same standards for assessing the quality of a segmentation. Furthermore, the results can depend on the order in which the human operator observes them. To minimize bias, such a method requires a large set of objects and a large group of humans. Unfortunately, this kind of method cannot be integrated in an automatic system.
- *System level evaluation methods*. This kind of methods indicates if the characteristics of the results obtained by a segmentation algorithm are suited for the over-all system which uses this segmentation algorithm. However, this evaluation method is indirect. If the process which follows the segmentation generates better results, it does not necessarily mean that the segmentation results were superior, and vice versa.
- *Empirical goodness or unsupervised methods*. They evaluate the performance of the algorithms by judging the quality of the segmented images themselves. To achieve this task, a set of quality criteria has to be defined. These

criteria are established according to human intuition about what conditions should be satisfied by an ideal segmentation. However, it seems difficult to establish quantitatively the quality of a segmentation only by using such an a priori criteria.

– *Empirical discrepancy or supervised methods*. A set of reference images presenting the ideal segmentation is first of all built. This set of images, which can be manually segmented by experts of the domain, constitutes a ground-truth. The purpose is to measure the discrepancy between these reference segmentations and that obtained by an algorithm to evaluate. So, these methods try to determine how far a segmented image obtained by an algorithm is from one or several segmented images. A large discrepancy involves a large segmentation error and thus this indicates a low performance of the considered segmentation algorithm.

The empirical discrepancy methods are the most popular for 2D-image segmentation evaluation [21, 28]. Indeed they seem to be the most suited for a quantitative evaluation as the measures of quality can be numerically computed, and for an objective evaluation thanks to the ground-truth.

Martin et al. [21] have proposed such a method to evaluate image segmentation algorithms. They built a public corpus containing ground-truth segmentations produced by human volunteers for images of a wide variety of natural scenes. They also defined a measure of segmentation similarity based on the computation of refinement error of a pixel between two segments (i.e. regions) containing this pixel.

In the 3D-domain, there exist some works proposing the quality assessment of segmentation in a specific context. In the MRI (Magnetic Resonance Imaging) field for example, Gerig et al. [13] propose a tool that quantifies the segmentation quality of 3D-images (volumetric images) including different shape distance metrics such as maximum Hausdorff distance, and mean/median absolute distance between object surfaces. For texture mapping, Sander et al. [23] introduce a metric based on the texture stretch induced by the parametrization of the segmented regions and allowing for the evaluation of the segmentation quality.

More recently, Attene et al. [3] have proposed some criteria like the aspect of the boundaries (smoothness, length), the hierarchical/multi-scale properties, the robustness, the complexity and the number of parameters. However, these criteria rather judge some technical points than the *real* quality of the techniques themselves, they rather fall in the empirical goodness methods. As raised by the authors, the main problem is that the objective quality of a segmentation of a given model is quite difficult to define, since it depends on the viewer's point of view and knowledge.

Berretti et al. [5] have presented some experimental results which are based on a ground-truth to validate their own segmentation algorithm. However, the ground-truth is not available on-line and according to the authors it contains very simple 3D-models (surfaces of revolution, vases, etc.).

Lastly, we proposed a framework to study the assessment problem of 3D-mesh segmentation [4]. Another work proposed by Chen et al. [7] addresses the same task. Both of these works propose a benchmark which is based on a ground-truth corpus of human segmented 3D-models, so they both constitute empirical discrepancy methods; the evaluation of a segmentation algorithm is realized by quantifying the consistency between the reference segmentations of the ground-truth corpus and those obtained by this algorithm on the same models using a set of similarity metrics that we will detail in the next subsection.

2.2 Review and analytic study of mesh segmentation similarity metrics

In the following, we summarize the existing metrics used to evaluate 3D-mesh segmentation and check if they are really reliable in the context of 3D-mesh segmentation evaluation. A reliable measure of mesh segmentation similarity has to possess the following set of properties:

– *No degenerative cases*. The score's measure must be proportional to the similarity degree between an automatic segmentation and the ground-truth segmentations of the same model. For example, an over-segmentation where each vertex (or face) is represented by a segment must give a very low value of similarity, since no ground-truth segmentation can be represented in such a way.
– *Tolerance to refinement*. The segmentation performed by some human observers can be coarse while the segmentation performed by others can be finer. However, they basically remain consistent; the difference just lies in the level of refinement. Hence, a reliable segmentation measure has to accommodate and to be invariant to these segmentation granularity differences.
– *Cardinality independence*. The measure must neither assume equal cardinality nor depend on this attribute. This means that two segmentations to be compared can have different numbers of segments and different sizes of segments.
– *Tolerance to cut boundary imprecision*. The segment boundaries are defined in a subjective way. Indeed, it is possible that two volunteers define the same segment on a model with a slight difference between boundaries, however, from a semantic point of view, the segments remain similar. Hence, a reliable measure has to accommodate this imprecision of cut boundaries.
– *Multiple ground-truth*. The measure has to be able to compare one automatic segmentation with multiple ground-truth (reference segmentations) for a given model, otherwise, providing multiple ground-truth in a benchmark is useless. An alternative solution is to simply average the similarity scores obtained between an automatic

segmentation and each manual segmentation (reference segmentation), however, this may bias the result and not really reflect how much an automatic segmentation agrees with the multiple ground-truth.
– *Meaningful comparison*. The scores obtained by the measure have to allow for a meaningful comparison between different segmentations of the same model and between segmentations of different models. For the first case (segmentations of the same model), the scores have to vary according to the segmentation quality, then, more the automatic segmentation is similar to the ground-truth segmentations of the same model, and better the score is. For the second case (segmentations of different models), the scores have to indicate which kind of 3D-models is the most convenient to segment by an automatic algorithm.

Essentially, the measures used to evaluate 3D-mesh segmentation can be classified into three categories: boundary matching, region differencing and non-parametric tests based measures.

In order to be able to formulate the above measures, we need to define what is a mesh segmentation. We will use this definition (according to Shamir [24]) for the remainder of this article.

Definition 1 Let M be a 3D-mesh, and R the set of mesh elements which are the vertices v_i or the faces f_i of M. A segmentation S of M is the set of sub-meshes $S = \{M_0, \ldots, M_{k-1}\}$ induced by the partitioning of R into k disjoint sub-sets of vertices or faces.

The three categories of measure are:

1. *Boundary matching*. This kind of measures compute the mapping degree between the extracted region boundaries of two segmentations. Chen et al. [7] proposed to use such a measure called *Cut discrepancy*. It measures the distances between cuts, where each cut represents an extracted region boundary. Let S_1 and S_2 be two segmentations of a 3D-mesh M and C_1, C_2, their respective sets of points on the segment boundaries. Let $d_G(p_1, C_2) = \min\{d_G(p_1, p_2), \forall p_2 \in C_2\}$ be the geodesic distance from a point $p_1 \in C_1$ to a set of cuts C_2.

   The Cut discrepancy between S_1 and S_2 is then:

$$\mathrm{CD}(S_1, S_2) = \frac{\mathrm{DCD}(S_1 \Rightarrow S_2) + \mathrm{DCD}(S_2 \Rightarrow S_1)}{\mathrm{avgRadius}}$$

   where, avgRadius is the average Euclidean distance from a point on the surface to centroid of the mesh, and DCD is a directional function defined as $\mathrm{DCD}(S_1 \Rightarrow S_2) = \mathrm{mean}\{d_G(p_1, C_2), \forall p_1 \in C_1\}$.

   A value of 0 will indicates a perfect matching between S_1 and S_2. As observed by Chen et al. [7] the measure is undefined when the model has no cuts and decreases to zero as more cuts are added to a segmentation. Hence it suffers from a degenerative case (see Sect. 2.2). In addition, it is not tolerant to refinement since for two segmentations that are perfect mutual refinements of each other, it can provide a large value. Moreover, for the unmatched points, it is possible to change their locations randomly and the measure will keep the same value. It is also not tolerant to imprecision of cut boundaries since it is based on a geodesic distance. Finally, it allows one to compare an automatic segmentation to only one ground-truth segmentation.
2. *Region differencing*. These measures compute the consistency degree between the regions produced by two segmentations S_1 and S_2. Berretti et al. [5] have proposed an overlap index representing the extent to which a region R_i of an automatic segmentation overlaps to closest region R_j of a ground-truth segmentation. The overlap index O_{index} of R_i is defined as:

$$O_{\mathrm{index}} = \max_j \frac{A(R_i \cap R_j)}{A(R_i)}$$

   with $A(.)$ the operator that returns the area of a region. If we suppose that S_1 is the automatic segmentation and S_2 is the ground-truth segmentation, then the distance between them is the average of the Overlap index overall regions of S_1. This measure falls in a degenerative case when R_i is represented by one face. Then the over-partitioning is not captured and it also does not allow for a comparison to multiple ground-truth.

   We (Benhabiles et al. [4]) and Chen et al. [7] proposed to use the consistency error measure. It is based on the computing of a local refinement error L_{3D} of a vertex (or face) v_i between S_1 and S_2 and is defined as:

$$L_{3D}(S_1, S_2, v_i) = \frac{|R(S_1, v_i) \backslash R(S_2, v_i)|}{|R(S_1, v_i)|}$$

   where the operator\denotes the set differencing, $|x|$ the cardinality of the set x, and $R(S, v_i)$ the region in segmentation S that contains the vertex v_i, i.e. the subset of vertices corresponding to a sub-mesh M_j of S containing v_i.

   This local refinement error produces a positive real valued output that presents the ratio of the number of vertices not shared between the first segment and the second one.

   Given this L_{3D}, there exist two ways to combine it for all vertices into a global measure for the entire 3D-mesh: the Global Consistency Error (GCE) and the Local Consistency Error (LCE).

The Global Consistency Error (GCE) forces all local refinements to be in the same direction and is defined as:

$$\mathrm{GCE}(S_1, S_2) = \frac{1}{N} \min\left\{\sum_i L_{3D}(S_1, S_2, v_i), \sum_i L_{3D}(S_2, S_1, v_i)\right\}$$

The Local Consistency Error (LCE) allows for different directions of refinement in different segments of the 3D-mesh:

$$\mathrm{LCE}(S_1, S_2) = \frac{1}{N} \sum_i \min\left\{L_{3D}(S_1, S_2, v_i), L_{3D}(S_2, S_1, v_i)\right\}$$

where N is the number of vertices. For both the GCE and the LCE, a value of 0 indicates a complete similarity, whereas a value of 1 indicates a maximum deviation between the two segmentations being compared. There are two degenerative segmentations that achieve a GCE and a LCE score of zero: one vertex per segment, and one segment for the entire mesh. We can also notice that the measure does not allow for a comparison to multiple ground-truth.

Chen et al. [7] proposed to use another measure namely Hamming distance. The Hamming distance between two segmentations S_1 and S_2 measures the region differencing between their respective set of segments. The directional Hamming distance is defined as:

$$D_H(S_1 \Rightarrow S_2) = \sum_i \left\| R_2^i \backslash R_1^{i_t} \right\|$$

where the operator$\backslash$denotes the set differencing, $\|x\|$ the cardinality of the set x, and $i_t = \mathrm{argmax}_k \|R_2^i \cap R_1^k\|$ which allows one to find the closest segment in S_1 to the region (or segment) R_2^i in S_2.

Given this D_H, and considering S_2 as the ground-truth, the authors of [7] defined the missing rate M_r and the false alarm rate F_r as follows:

$$M_r(S_1, S_2) = \frac{D_H(S_1 \Rightarrow S_2)}{\|S\|}$$

$$F_r(S_1, S_2) = \frac{D_H(S_2 \Rightarrow S_1)}{\|S\|}$$

and the Hamming distance as the average of missing rate and false alarm rate:

$$\mathrm{HD}(S_1, S_2) = \frac{1}{2}\big(M_r(S_1, S_2) + F_r(S_1, S_2)\big)$$

As observed by the authors [7] the measure has a good behavior when the correspondences between segments are correct but it fails when they are not. Another limit is the comparison to only one ground-truth.

3. *Non-parametric tests*. In the statistical literature there exists a lot of non-parametric measures. We can cite for example Cohen's Kappa [8], Jaccard's index [11], Fowlkes and Mallow's index [11]. The latter two are variants of Rand index [22]. Chen et al. [7] proposed to use Rand index for 3D-mesh segmentation evaluation. This index converts the problem of comparing two segmentations S_1 and S_2 with different numbers of segments into a problem of computing pairwise label relationships. If we denote $l^i_{S_1}$ the corresponding label of all elements (vertices or faces) contained in region R_i of S_1 and similarly $l^i_{S_2}$ the corresponding label of all elements contained in region R_i of S_2, the Rand index can be computed as the ratio of the number of pairs of vertices or faces having the compatible label relationship in S_1 and S_2 and can be defined as:

$$\mathrm{RI}(S_1, S_2) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} \mathbf{I}\big(l^i_{S_1} = l^j_{S_1}\big)\big(l^i_{S_2} = l^j_{S_2}\big) + \mathbf{I}\big(l^i_{S_1} \neq l^j_{S_1}\big)\big(l^i_{S_2} \neq l^j_{S_2}\big)$$

where $\mathbf{I}$ is the identity function, and the denominator is the number of possible unique pairs among N vertices or faces. This gives a measure of similarity ranging from 1, when the two segmentations are identical, to 0 otherwise. This measure does not allow for comparison to multiple ground-truth segmentations.

We can notice that all existing measures suffer from either degenerative cases and/or sensitivity to refinement and/or sensitivity to cut boundaries imprecision and/or limitation in term of comparison to multiple reference (i.e. ground-truth) segmentations. Therefore none of these measures satisfies the whole set of defined criteria.

3 The 3D probabilistic rand index (3D-PRI)

The goal of this measure is to perform a quantitative comparison between a mesh segmentation algorithm and a set of ground-truth segmentations (of the same model). In the field of 2D-image, Unnikrishnan et al. [28] proposed a probabilistic interpretation of Rand Index to evaluate the performance of 2D-image segmentation algorithms and shown the relevance of the obtained results. Hence we have generalized this measure for 3D-mesh segmentation evaluation.

Let S_a be the automatic segmentation to be compared to a set of manual segmentations (ground-truth) $\{S_1, S_2, \ldots, S_K\}$ of a 3D-mesh M. We denote the corresponding label of a vertex v_i (label of the segment to which belongs vertex v_i) by $l^i_{S_a}$ in segmentation S_a and by $l^i_{S_k}$ in the ground-truth segmentation S_k. It is assumed that the label $l^i_{S_k}$ takes a value ranged between 1 and the number of segments of S_k and

similarly $l^i_{S_a}$ takes a value ranged between 1 and the number of segments of S_a. The label relationships for each vertex pair is modeled by an unknown underlying distribution. This can be considered as a process where each human segmenter provides information about the segmentation S_k of the 3D-mesh in the form of binary numbers $\mathbf{I}(l^i_{s_k} = l^j_{s_k})$ for each pair of vertices (x_i, x_j). The set of all perceptually correct segmentations defines a Bernoulli distribution over this number, giving a random variable with expected value denoted as p_{ij}. Hence, the set $\{p_{ij}\}$ for all unordered pairs (i, j) defines a generative model of correct segmentations for the 3D-mesh M. The 3D Probabilistic Rand Index is then defined as:

$$\text{3DPRI}\big(S_a, \{S_k\}\big) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} e_{ij} p_{ij} + (1 - e_{ij})(1 - p_{ij}) \quad (1)$$

where e_{ij} denotes the event of a pair of vertices i and j belonging to the same segment (or region) in the automatic segmentation:

$$e_{ij} = \mathbf{I}\big(l^i_{S_a} = l^j_{S_a}\big)$$

and p_{ij} denotes the probability of the vertices i and j belonging to the same segment in the ground-truth set $\{S_k\}$ and is given by the sample mean of the corresponding Bernoulli distribution as suggested by Unnikrishnan et al. [28]:

$$p_{ij} = \frac{1}{K} \sum_k \mathbf{I}\big(l^i_{S_k} = l^j_{S_k}\big)$$

The 3D-PRI takes a value ranged between 0 and 1, where 0 indicates no similarity between S_a and $\{S_1, S_2, \ldots, S_k\}$, and 1 indicates a perfect similarity.

Note that with this formulation for p_{ij}, computing the 3D-PRI is equivalent to averaging the RI over the multiple ground-truths. However, the 3D-PRI formulation is generic and we can imagine a different and more efficient way to compute the p_{ij}. The main advantage of the simple mean estimator is its fast computation.

We have noticed in practice, however, that the 3D-PRI suffers from lack of discriminative power in its values. Indeed, the values obtained by the index do not allow one to clearly decide if a segmentation obtained by an automatic algorithm is relevant or not. This is due to the limited effective range of 3D-PRI in term of maximum and minimum value. To address this drawback, we present in the next section, the 3D normalized probabilistic Rand index (3D-NPRI).

4 3D normalized probabilistic rand index (3D-NPRI)

Our objective is to normalize the 3D-PRI, in order to increase its dynamic range and thus its discriminative power. Hence we need to define a baseline to which the index can be expressed. For 3D-mesh segmentations, the baseline may be interpreted as the expected value of the index under some particular segmentations of the input 3D-model. A popular strategy [11, 28] of index normalization with respect to its baseline is:

$$\text{Normalized index} = \frac{\text{Index} - \text{Expected index}}{\text{Maximum index} - \text{Expected index}} \quad (2)$$

As observed by Unnikrishnan et al. [28] there is a little agreement in the statistics community regarding whether the value of "Maximum index" should be estimated from the data or set constant. We choose to follow what was done by Unnikrishnan et al. [28] and set the value to be 1 (the maximum possible value of the 3D-PRI). Thus, we avoid the practical difficulty of estimating this quantity for complex data sets.

Another parameter to define is the expected probabilistic Rand index E(3D-PRI). One may draw an analogy between the E(3D-PRI) and the 3D-PRI in equation 1 as follows:

$$E\big[\text{3DPRI}\big(S_a, \{S_k\}\big)\big] = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} e'_{ij} p_{ij} + (1 - e'_{ij})(1 - p_{ij}) \quad (3)$$

where $e'_{ij} = E[\mathbf{I}(l^i_{S_a} = l^j_{S_a})]$. This latter quantity has to be computed in a meaningful way. Unnikrishnan et al. [28] proposed to estimate it from segmentations of all images of the database for all unordered pairs (i, j). Let Φ be a number of images in a data set and K_ϕ the number of ground-truth segmentations of image ϕ. Then, e'_{ij} is expressed as:

$$e'_{ij} = \frac{1}{\Phi} \sum_\phi \frac{1}{K_\phi} \sum_{k=1}^{K_\phi} \mathbf{I}\big(l^i_{S^k_\phi} = l^j_{S^k_\phi}\big)$$

However, this estimation can only be used in a data-base of 2D-images having equal sizes (where each pixel has its correspondent over all the other segmented images). In the 3D case, it is not possible, since the different models of the corpus have different number of vertices and different connectivities. One possible way to compute the E(3D-PRI) while keeping a correct baseline and without having any constraint on the corpus, is to use random segmentations S_r:

$$E\big[\text{3DPRI}\big(S_a, \{S_k\}\big)\big] = \frac{1}{N} \sum_{r=1}^{N} \text{3DPRI}\big(S_r, \{S_{K_r}\}\big) \quad (4)$$

where N is the number of 3D-models in our corpus and $\{S_{k_r}\}$ are ground-truths of the model concerned by S_r. We then

Fig. 1 Random segmentations of some 3D-models of the corpus

define the 3D-NPRI of an automatic segmentation of a given 3D-model as follows:

$$\text{3DNPRI}(S_a) = \frac{\text{3DPRI}(S_a, \{S_K\}) - E[\text{3DPRI}(S_a, \{S_k\})]}{1 - E[\text{3DPRI}(S_a, \{S_k\})]} \tag{5}$$

The random segmentations were generated using a simple algorithm: L seed vertices were randomly chosen on the object, then L connected regions were obtained by a simple region growing mechanism. The number of segments (or regions) takes a value ranged between 2 and the number of vertices of the concerned model. Figure 1 shows some 3D-models of the corpus on which the random segmentation algorithm was applied. We have to precise here that the 3D-NPRI is not affected by the choice of these random segmentations. Indeed we will show later (see Fig. 3) that the 3D-PRI provides very stable values when comparing ground-truth segmentations to random segmentations (even with very different granularities) hence the normalization constant E(3D-PRI) (see (4)) is almost invariant to the choice of the random segmentations S_r.

Hence, the 3D-NPRI will take a value with a lower bound of -1 and an upper bound of 1, where -1 indicates no similarity between the automatic segmentation and the ground-truth segmentations of the same model, and 1 indicates a perfect match. The lower bound of -1 is explained by the fact that the expected Index cannot exceed 0.5 since we compare a set of random segmentations to a set of ground-truth segmentations (see Sect. 6.1). Therefore, the worst case will be:

$$\text{3DNPRI}(S_a) = \frac{0 - 0.5}{1 - 0.5} = -1$$

where the automatic segmentation has no similarity with its corresponding ground-truths.

Note that the metric's definition does not take into account model with different sampling. Moreover, the score of the metric changes by changing the order of vertices on the automatic segmentation and the ground-truths of the same model. However, in our case, it is not really a drawback since we compare segmentations of the same model while keeping the same sampling and the same order of vertices.

5 Ground-truth corpus

The current version of our corpus is an improved version of [4] in term of number of models and ground-truth segmentations per model. The corpus is available on-line[2] and contains twenty-eight 3D-models (as triangle meshes) grouped in five classes, namely *animal*, *furniture*, *hand*, *human* and *bust*. Each 3D-model of the corpus is associated with 4 manual segmentations which give a total of 112 ground-truth segmentations done by 36 volunteers. Figure 2 illustrates the models of the corpus with one manual segmentation per model. We have selected a small number of varied models with respect to a set of properties. All the selected models are manifold, connected, and do not have intersecting faces. Hence they are supported as an input by any segmentation algorithm. In order to collect precise manual segmentations, we have assisted the volunteers in tracing the vertex-boundaries through the different models. Note that the volunteers have freely segmented the models and no condition was imposed on the manner with which they have segmented them. For this task, we used MeshLab[3] application allowing for an explicit vertex-per-vertex segmentation of models using colors.

Chen et al. [7] proposed another corpus that seems complementary to ours: they present more objects (380 3D-models of the Watertight Track of the 2007 SHREC Shape-based Retrieval Contest [14]) when we selected a small representative set (it allows one to rapidly evaluate a segmentation algorithm without running it on 380 objects). They chose to use the web application Amazon's Mechanical Turk[4] to collect the manual—i.e. ground-truth—segmentations without any supervision when we chose to supervise our volunteers to obtain more precise manual segmentations. Finally, their ground-truth presents face-based segmentations whereas ours contains vertex-based segmentations.

6 Experimental comparison of properties of existing segmentation similarity metrics

In what follows, we provide an experimental study of the 3D-PRI/3D-NPRI properties and we compare them to the existing metrics for assessing 3D-mesh segmentation quality. For this end, we use our corpus models and their corresponding ground-truths.

Most of the measures introduced in Sect. 2.2 quantify *dissimilarity* (the lower is the number, the better is the segmentation result) between segmentations rather than *similarity*. In order to have a meaningful comparison between

[2] http://www-rech.telecom-lille1.eu/3dsegbenchmark/.

[3] http://meshlab.sourceforge.net/.

[4] http://www.mturk.com/.

Fig. 2 Models of our corpus associated with one ground-truth

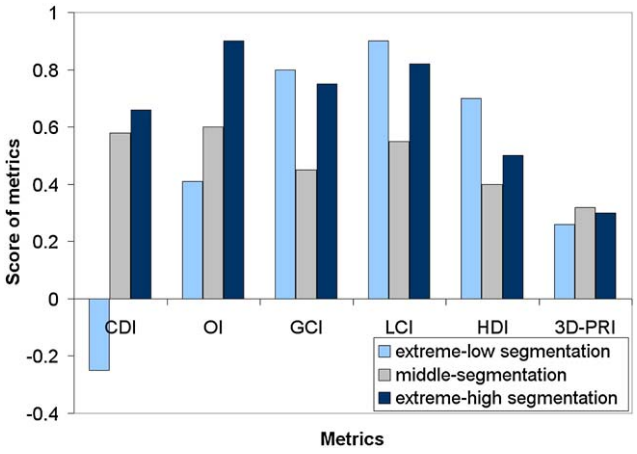


Fig. 3 Comparison of three levels of random segmentation (extreme-low, middle, and extreme-high) to the ground-truths for the whole corpus using different indexes

these measures and the 3D-PRI/3D-NPRI, we define the quantities $\mathrm{CDI}(S_1, S_2) = 1 - \mathrm{CD}(S_1, S_2)$, $\mathrm{GCI}(S_1, S_2) = 1 - \mathrm{GCE}(S_1, S_2)$, $\mathrm{LCI}(S_1, S_2) = 1 - \mathrm{LCE}(S_1, S_2)$, and $\mathrm{HDI}(S_1, S_2) = 1 - \mathrm{HD}(S_1, S_2)$. The "I" in the acronyms stands for "Index", complying with the popular usage of the term in statistics when quantifying similarity. Hence, except the CDI, all of the other indexes are in the range $[0, 1]$ with a value of 0 indicating no similarity between segmentations of the same model and a value of 1 indicating a perfect match. The CDI is in the range $]-\infty, 1]$.

6.1 Sensitivity to degenerative cases

The first property to study is the sensitivity of each index regarding degenerative cases. For this end, we compare our Probabilistic Rand Index (3D-PRI) with the Cut Discrepancy Index (CDI), the Hamming Distance Index (HDI), the Global and Local Consistency Index (GCI/LCI), and the Overlap Index (OI) for three kinds of random segmentations namely *extreme-low segmentation* (segmentation composed of a 2 or 3 segments), middle-segmentation (segmentation composed of a number of segments which is similar to that of ground-truths of the corresponding model), and *extreme-high segmentation* (segmentation composed of more than 50 segments). They were generated using a random segmentation algorithm. Figure 3 presents the results obtained by the comparison of these random segmentations to the set of the ground-truths for each model of the corpus. Each index of the figure is computed for the three kinds of segmentation (extreme-high segmentation, middle-segmentation, and extreme-low segmentation) and averaged across the entire data set. Since the segmentations are random, the scores obtained by the metrics are expected to be low for the three kinds of segmentation, and it is the case for the 3D-PRI. We can notice, however, that although the random segmentations are totally different from the ground-truths, the scores of the other metrics are very high (very good) for certain segmentations with degenerative granularity (extreme-high and/or extreme-low). Hence the 3D-PRI is the most stable regarding degenerative cases considering its scores, which are less than 0.32.

6.2 Tolerance to refinement

The second property to study is the tolerance of each index to refinement. For this end, we perform two kinds of experiments. The first one uses segmentations with mutual refinements, and the second one uses segmentations with hierarchical refinements. The obtained results for the first experiment are presented in Fig. 4.

It shows two segmentations of the *dinopet* model which are perfect mutual refinements of each other, and a plot in which is computed the similarity between the two segmentations using different metrics. The plot of Fig. 4 clearly shows that the CDI fails to capture the similarity between the two segmentations (a) and (b). Although the two segmentations are similar (the difference just lies in the level of refinement). However, the other metrics have a good behavior toward this kind of refinement since all of them give scores which are close to 1.

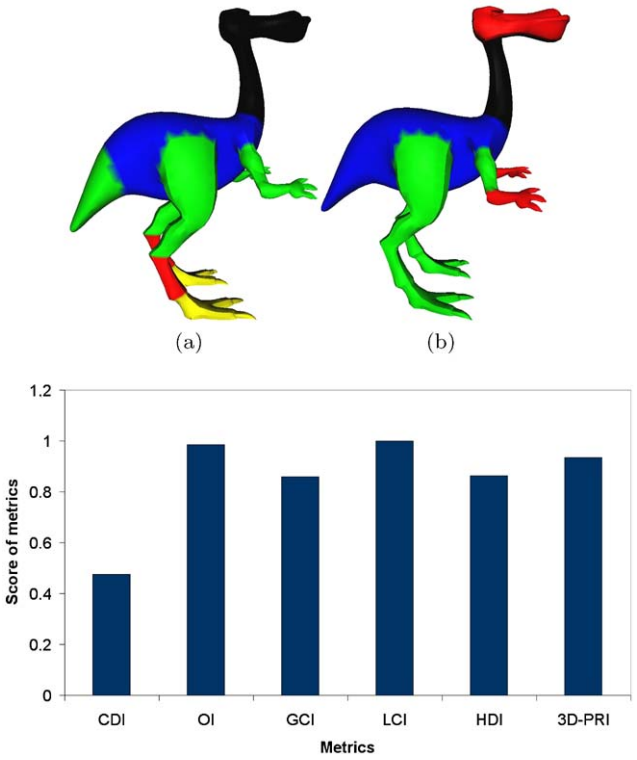


Fig. 4 Tolerance to mutual refinement of different indexes, by comparing two segmentations (**a**), (**b**) with perfect mutual refinement for the dinopet model

The second experiment was performed using the hierarchical segmentation algorithm of Attene et al. [2]. We generated several levels of segmentation (from 4 segments to 15 segments) on the *horse* model of our corpus then we compared these 12 versions to the ground-truths. Figure 5 illustrates the obtained results using different indexes. The OI and the GCI does not appear on the figure since they have the same behavior as the LCI. The figure clearly shows that the CDI is less stable toward hierarchical refinement than the other indexes. The LCI seems completely invariant while the 3D-PRI and the HDI present a slight variation; they are not fully invariant but present a good tolerance to refinement.

6.3 Independence from cardinality

The third property to study is the independence of each index toward segmentation cardinality. According to the previous performed experiments about the first two properties (degenerative cases and refinement), the CDI seems to be the only metric which depends on the cardinality, in a critical way. Indeed, the comparison between two segmentations with different number of segments will give a bad score using this metric whatever the quality.

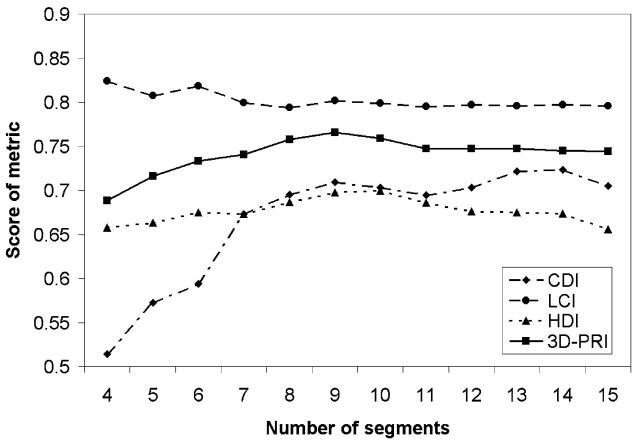


Fig. 5 Tolerance to hierarchical refinement of different indexes, by comparing several levels of segmentation of the horse model to its corresponding ground-truths

6.4 Tolerance to imprecision of cut boundaries

The fourth property to study is the tolerance of each index to the imprecision of cut boundaries. For this end, we manually segmented a simple model (bitorus) into 2 segments. We proposed 5 segmentations (Fig. 6(a) to (e)) where each one of them has a slight difference in the boundary position with comparison to the others, then we computed the similarity between segmentation (c) and the other segmentations. The plot in Fig. 6 shows the obtained results using different indexes. Contrary to the other indexes, the CDI gives low values of similarity between segmentations. Although the CDI is not in the same range as the other metrics, the plot still allows us to illustrate the qualitative behavior of this latter index toward the imprecision of cut boundaries. We can notice also that except the 3D-PRI which presents a slight variation but a good tolerance, the other indexes are almost invariant.

At this point, we have shown that the 3D-PRI satisfies the five properties: ability to compare one automatic segmentation with multiple ground-truth segmentations, no degenerative cases, tolerance to refinement, independence from segmentation cardinality, and tolerance to imprecision of cut boundaries. We also have shown that the 3D-PRI outperforms the other indexes in terms of the first two properties. We show in the next experiments that the normalization of this index (into 3D-NPRI) improves its discriminative power and give better results in term of meaningful comparison.

6.5 Meaningful comparison

The main advantage of the 3D-NPRI is the ability to provide values that allow a meaningful comparison between segmentations of different 3D-models. Figure 7 demonstrates this behavior. The top two rows show different 3D-models of our corpus segmented at different granularity using the

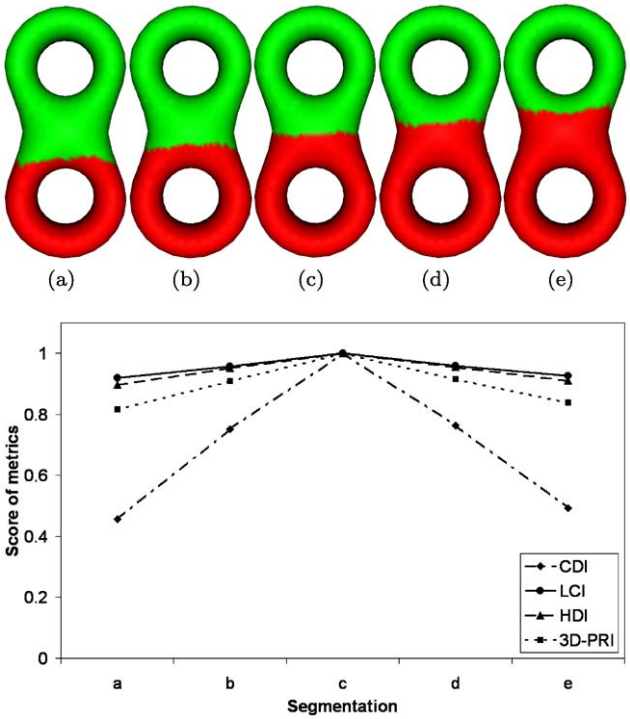


Fig. 6 Tolerance to imprecision of cut boundaries of different indexes, by comparing segmentation (**c**) to segmentations (**a**) to (**e**) for the bitorus model

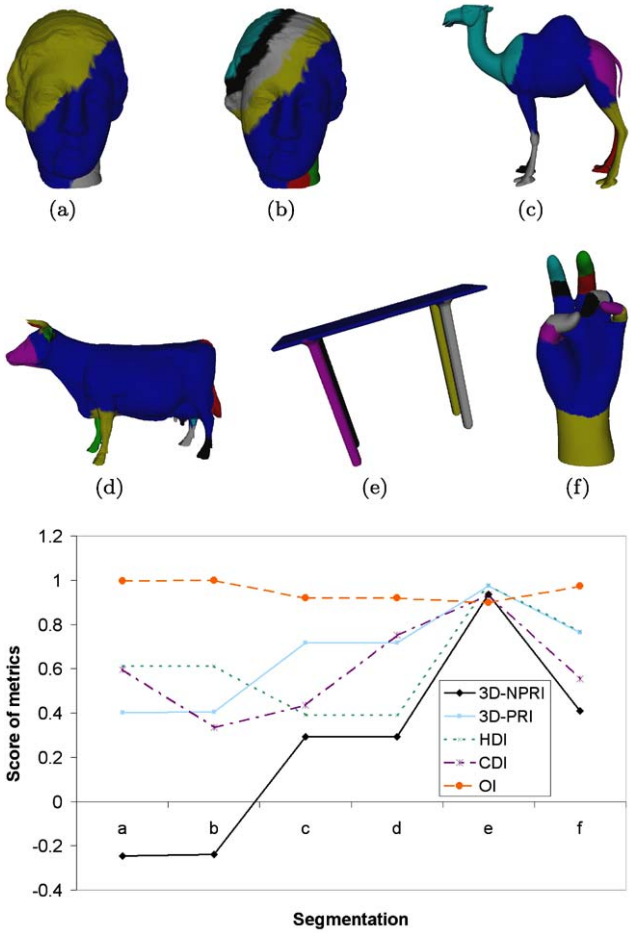


Fig. 7 Example of comparing segmentations of different models: From (**a**) to (**f**) segmentations using algorithm from [27]. The plot (**g**) shows the scores of different indexes for each segmentation (**a**) to (**f**)

hierarchical algorithm of Tierny et al. [27]. These automatic segmentations are compared to the ground-truth corpus (see Fig. 2) using the previous indexes and our 3D-NPRI. Visually, regarding the ground-truth, segmentations (a) and (b) (Fig. 7) seem very poor, segmentations (c), (d), and (f) are correct, and segmentation e is perfect. One can notice that the OI similarity is high for all of the 3D-models. Hence, it cannot indicate which segmentation is the best. Note that although the HDI gives lower scores than the OI, it also fails to distinguish between correct and poor segmentations since it gives high values for poor ones (Fig. 7(a) and (b) and low values for correct ones (Fig. 7(c) and (d). The GCI/LCI does not appear in the plot in order to keep a clear display. This latter metric has the same behavior than HDI. The CDI has slightly a better behavior than HDI but still to fail distinguishing between correct and poor segmentations. The 3D-PRI reflects the correct relationship among the segmentations. However, its range is small, and the expected value is unknown, hence it is difficult to determine which segmentation is really good. The 3D-NPRI fixes all of these drawbacks. It reflects the desired relationship among the segmentations with no degenerate cases. Besides, any segmentation which gives a score significantly above 0 can be considered as relevant (since it provides results significantly better than random segmentations).

6.6 Discriminative power

The best way to attest the discriminative power of our 3D-NPRI is to show that its values are well correlated with the rates given by users for a set of segmentations.

In the following experiment, we study the correlation between the metrics' values and the observers' rates of a set of segmentations obtained from our corpus models. For this end, we used the following algorithms: Attene et al. [2], Lavoué et al. [20], Shapira et al. [25], and Tierny et al [27]. We collected 250 segmentations including 28 ground-truth segmentations and 28 random segmentations. Except for the algorithm of Lavoué et al. [20], the others are hierarchical. Hence, we generated for each one of them two levels of segmentation, namely coarse and fine, which gives 28×2 segmentations per algorithm and 28 segmentations from Lavoué et al.'s [20] algorithm. We computed the quality index of these 250 segmentations (using our ground-truth) using the different metrics. We then asked several observers to give a rate reflecting the perceived quality of each segmen-

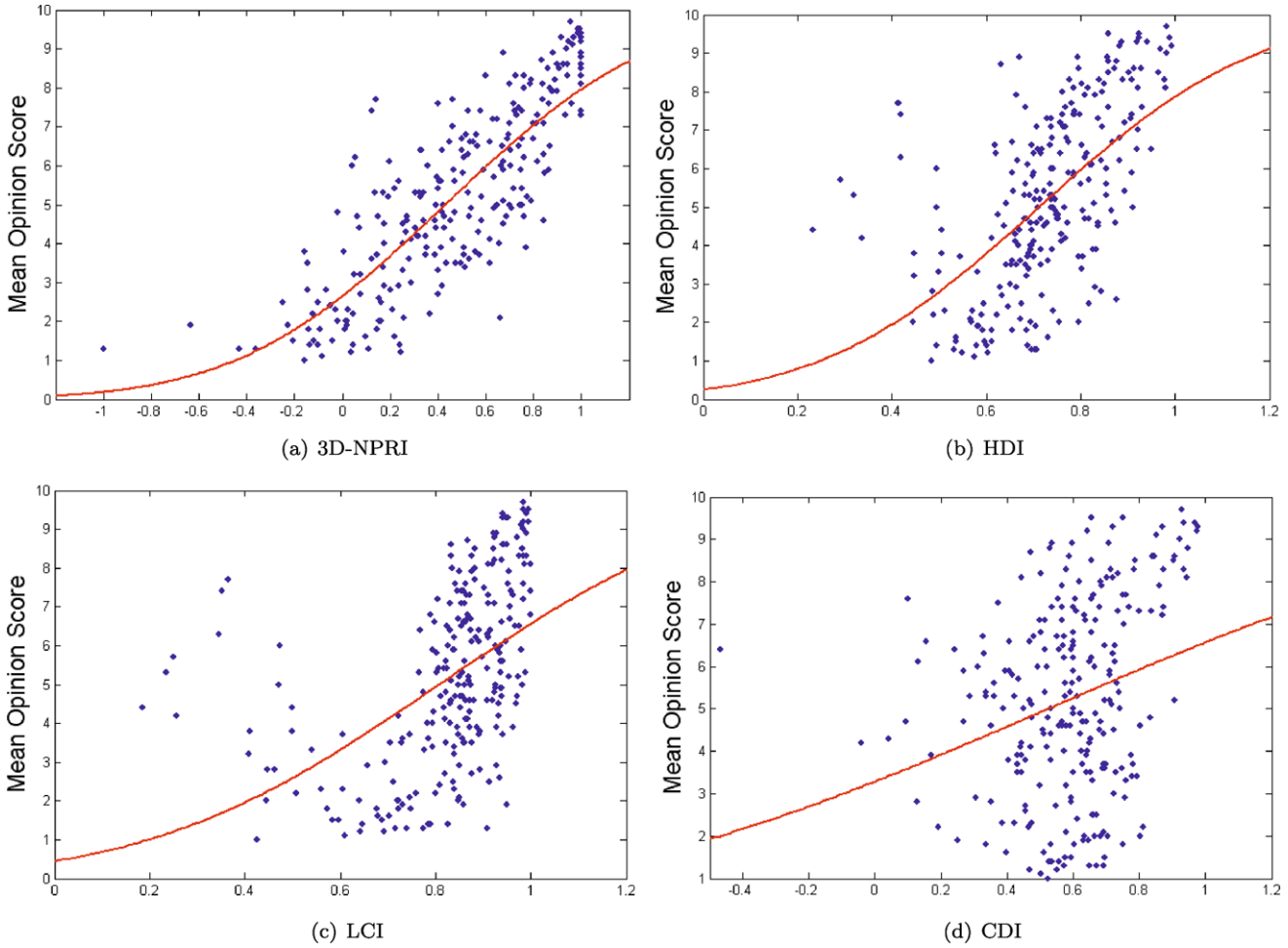


Fig. 8 Subjective MOS vs metric values for the whole corpus models and for different metrics. Each circle represents a segmentation. The Gaussian fitted curve is displayed in *red*

tation between 1 (bad segmentation) and 10 (perfect segmentation). Hence each segmentation was associated with quality index values from the different metrics and a subjective Mean Opinion Score (MOS) from human observers. This latter score reflects the opinion of observers toward the quality of a segmentation. The MOS of the segmentation i is computed as follows:

$$\mathrm{MOS}_i = \frac{1}{N}\sum_{j=1}^{N} m_{ij}$$

where N is the number of observers (10 in our experiment), and m_{ij} is the rate (in the range [1, 10], 10 for a very good segmentation) of the jth observer given to the ith segmentation. For the correlation, we considered a statistical indicator namely the Pearson Product Moment Correlation [10]. This indicator measures the linear dependence between two variables X and Y. In order to optimize the matching between the values of the different metrics and the MOS of observers, we performed a psychometric curve fitting using the Gaussian psychometric function (recommended by [9]).

Table 1 Pearson correlation values (%) between the Mean Opinion Scores and the values of different metrics for each model category of our corpus

	CDI	OI	GCI	LCI	HDI	3D-NPRI
Animal	2.6	2.3	9.3	8.3	16.9	58.7
Bust	10.9	0	45.9	61.1	54.8	77.4
Furniture	5.8	14.8	49.9	50.5	63	73.2
Hand	21.2	1	54.1	54.4	57.5	70.2
Human	1.5	5.5	32.1	32.6	39	51.6
Whole	7.1	2.6	23.7	20.9	32.9	66.1

Table 1 shows the results of correlation between the values of different metrics and the MOS of observers for Pearson indicator. The results in the table clearly shows that the 3D-NPRI outperforms the other metrics in term of correla-

Table 2 Properties of existing similarity metrics

	CDI	OI	GCI	LCI	HDI	3D-NPRI
Degenerative cases	Yes	Yes	Yes	Yes	Yes	No
Tolerance to refinement	No	Yes	Yes	Yes	Yes	Yes
Cardinality independence	No	Yes	Yes	Yes	Yes	Yes
Tolerance to cut imprecision	No	Yes	Yes	Yes	Yes	Yes
Multiple ground-truth	No	No	No	No	No	Yes
Meaningful comparison	No	No	No	No	No	Yes
Strong discriminative power	No	No	No	No	No	Yes

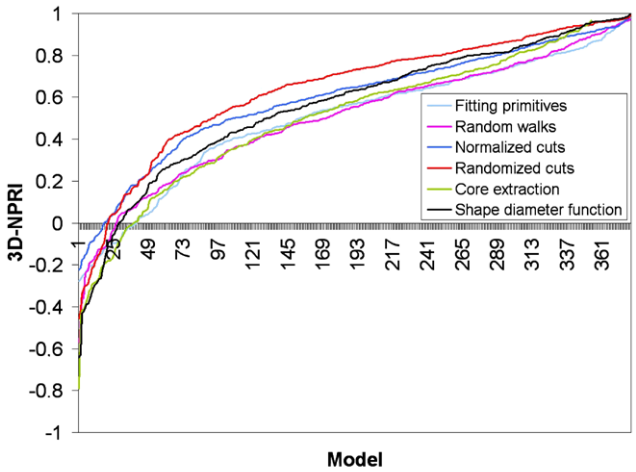
tion for each category and for the whole corpus. Moreover, the Pearson correlation value of the 3D-NPRI for the whole corpus is high (66.1%), when those of the other metrics are quite bad (less than 33%). This means that except the 3D-NPRI, the other metrics fail to distinguish between good and bad segmentations. Figure 8 presents the psychometric curve fitting between the objective and subjective scores for 3D-NPRI, HDI, LCI and CDI for 250 segmentations of the corpus models. It visually illustrates the superiority of the 3D-NPRI for predicting the subjective opinion, and leads to conclude that the 3D-NPRI has the best discriminative power. These results clearly validate the 3D-NPRI, since they are in agreement with the human opinion.

The properties of each metric are summarized in Table 2 according to the performed experiments in this section.

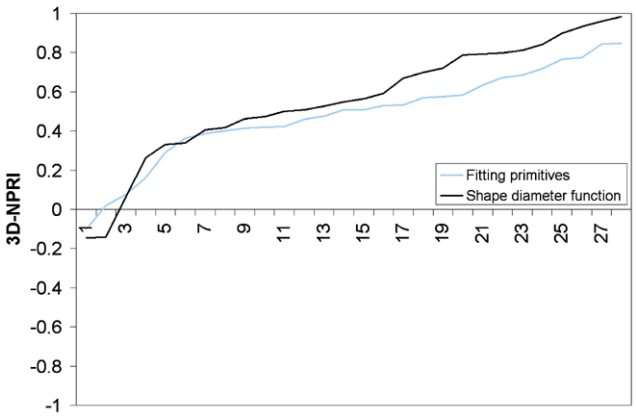
7 Application for the evaluation of recent segmentation algorithms

In this section, we apply the 3D-NPRI together with the Chen et al.'s [7] corpus and our corpus (described in Sect. 5) to evaluate a set of recent automatic segmentation algorithms, then we compare the obtained results by the two corpuses. We have considered the six recent automatic segmentation algorithms used in Chen et al. [7]: Attene et al. [2], Lai et al. [19], Golovinskiy et al. [15], Katz et al. [18], and Shapira et al. [25]. The six algorithms are respectively based on: fitting primitives, random walks, normalized cuts/randomized cuts, core extraction, and shape diameter function. The segmentations using these algorithms for the Chen's corpus are available on-line. On the other hand, we used Attene et al.'s [2], and Shapira et al.'s [25] algorithms (the only algorithms available on-line among the previous six) to generate automatic segmentations on our corpus models. The reader can refer to the original papers for more details about the six algorithms.

Note that all the algorithms cited above are part-type hierarchical segmentation methods. Hence for each one of them we can generate several levels of segmentation. Chen et al. [7] provided only one level of segmentation for each algorithm applied on their corpus. For this end, they used the parameter settings recommended by the authors of the algorithms. To keep a valid comparison between the two corpuses, we also used the parameter settings recommended by the authors of the algorithms to generate segmentations on our corpus models. Note that the level of segmentation will not influence the evaluation results since we proved that the 3D-NPRI is tolerant to hierarchical refinement (see Fig. 5).



(a) Results on Chen s et al. [7] corpus



(b) Results on our corpus

Fig. 9 Scores of 3D-NPRI sorted in increasing order over all the two corpus models

Table 3 Algorithms ranking applied on respectively the Chen's corpus and our corpus

Algorithm	3D-NPRI mean	Rank
Fitting primitives	0.49/0.49	5/2
Random walks	0.50/–	4/–
Normalized cuts	0.59/–	2/–
Randomized cuts	0.63/–	1/–
Core extraction	0.46/–	6/–
Shape diameter function	0.56/0.55	3/1

To ensure a relevant comparison between the algorithms, we compute the 3D-NPRI for every 3D-model of the Chen's corpus and of our corpus. Figure 9 shows the 3D-NPRI for each model of the two corpuses and for each algorithm. The values are sorted in increasing order for each algorithm, hence the jth model may not be the same across algorithms. This kind of graph was already applied for segmentation evaluation in the field of 2D-image [28].

Table 3 presents the rank of each algorithm together with the 3D-NPRI mean value over all the two corpuses.

Table 3 and Fig. 9 demonstrate, as expected, that the segmentations obtained by the six algorithms are relevant since most of the values of the 3D-NPRI are greater than zero. The Randomized Cut algorithm seems to provide the best results. It is very interesting to notice that the Fitting Primitives and Shape Diameter keep similar behavior for the two corpuses although these two corpuses are very different: the profiles of the 3D-NPRI distribution (see Fig. 9) and the mean 3D-NPRI values (see Table 3) for these algorithms are almost exactly the same for both corpuses. Hence it validates the fact that our corpus, since it presents high quality manual segmentation and heterogeneous models, is clearly efficient for segmentation evaluation despite its small size. Another interesting experiment is to see which category models the algorithms fail to segment well. For this end, we average the 3D-NPRI for each category of the two corpuses. Figures 10 and 11 illustrate the obtained results for the six algorithms. One can notice that whatever the corpus is, there is no algorithm that is reaching the highest scores for all categories. Moreover, each algorithm has at least one category inadequately segmented since its mean 3D-NPRI value is very low (close to 0 or less). The core extraction algorithm for instance fails to adequately segment the *Bearing* and *Mech* categories (see Fig. 11(e)). This result is straight since the concerned algorithm is a part-based one. Indeed, it tries to detect the core of a model which from a semantic point of view is hard to define in such categories. As observed by Chen et al. [7], some algorithms do not necessarily segment the best (with comparison to others) categories for which they were designed. We can notice this behavior on our corpus too. For instance, the algorithm based on Fitting Primitives gives greater 3D-NPRI score (better) for the *hand* category than the algorithm based on Shape Diameter Function and vice versa for the *furniture* category. As raised by Chen et al. [7], this means that either the human observers do not segment models in the expected way, or the part structures of these models are revealed by other properties.

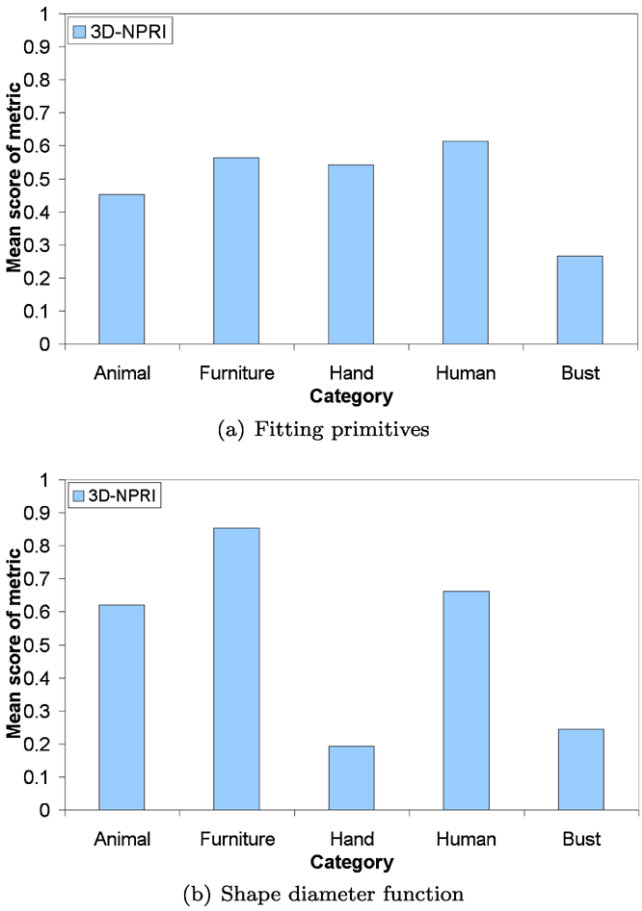


Fig. 10 Scores of 3D-NPRI averaged for each category models of our corpus

Our results and those of Chen et al. [7] are coherent. This is straight since our metric is a probabilistic interpretation of the Rand Index (metric used by Chen et al. [7] to analyze and evaluate the algorithms) to which we added a normalization allowing a better results analysis.

8 Conclusion

This paper presents a thorough comparison between existing similarity metrics and a new one addressing the assessment problem of mesh segmentation. For this end we use a corpus of manually segmented models. This corpus is an improved version of our previous one [4] and is available online. The new 3D-NPRI metric is a probabilistic interpretation of the Rand Index which allows to quantify the consis-

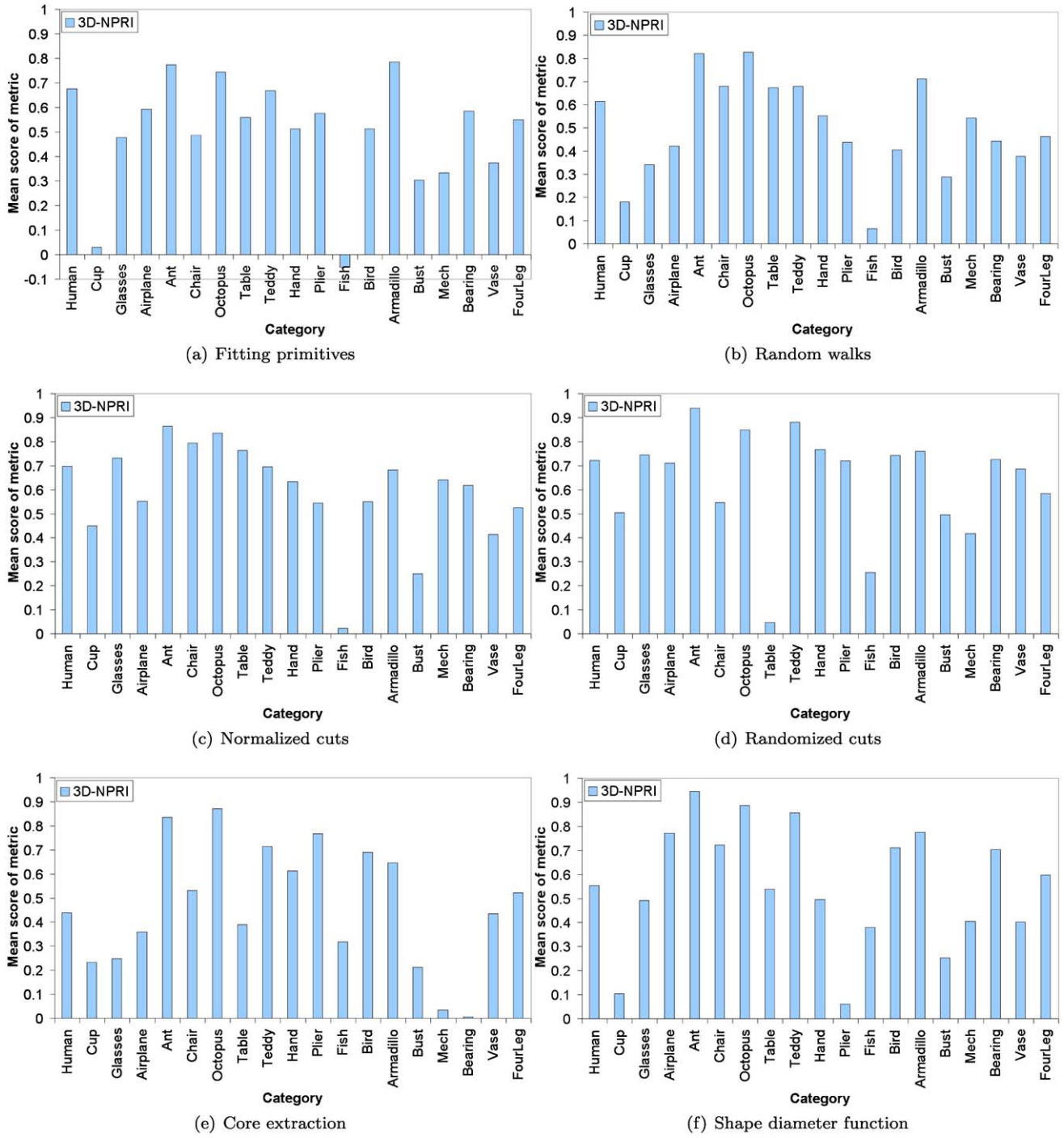


Fig. 11 Scores of 3D-NPRI averaged for each category models of the Chen's corpus

tency between multiple segmentations of a 3D-mesh model. The paper shows that this new metric outperforms existing ones in terms of properties and discriminative power. The results are validated by comparing subjective scores of human observers to the objective metric scores. Finally, the measure is applied together with the Chen et al.'s [7] corpus and our corpus to evaluate six recent 3D-mesh segmentation algorithms. This evaluation allowed to compare the obtained results depending on the corpus and showed their coherence.

For future work, we plan to explore other kind of estimator to compute the p_{ij} (see (1) in Sect. 3) in order to improve the correlation between metric's scores and the observes' scores, we also plan to enrich our subjective tests by integrating more experiments allowing to compare algo-

rithms. Finally, we plan to exploit our ground-truths to design a learning segmentation algorithm.

Acknowledgements We would like to thank Marco Attene, Ariel Shamir, Shy Shalom, and Julien Tierny for providing us the source code or the binary of the segmentation algorithms and Fatan Souhaib for implementing the 3dsegbenchmark site. We also would like to thank Xiaobai Chen for providing on-line the manual and automatic segmentations of different algorithms. We thank AIM@SHAPE, GAMMA-INRIA, and Princeton Shape Benchmark databases for providing 3D-mesh models.

This work is supported by the ANR (Agence Nationale de la Recherche, France) through MADRAS project (ANR-07-MDCO-015).

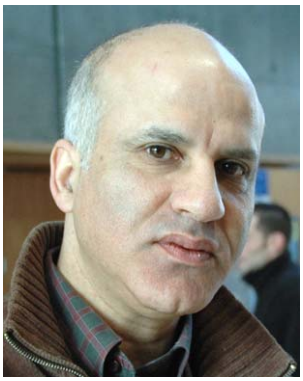
References

1. Antini, G., Berretti, S., Pala, P.: 3d mesh partitioning for retrieval by parts application. In: IEEE International Conference on Multimedia & Expo (ICMEÆ05) (2005)
2. Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives. Vis. Comput. **22**(3), 181–193 (2006)
3. Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., Tal, A.: Mesh segmentation, a comparative study. In: IEEE International Conference on Shape Modeling and Applications, p. 7 (2006)
4. Benhabiles, H., Vandeborre, J.P., Lavoué, G., Daoudi, M.: A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3d-models. In: IEEE International Conference on Shape Modeling and Application (SMI) (2009)
5. Berretti, S., Bimbo, A.D., Pala, P.: 3d mesh decomposition using Reeb graphs. Image Vis. Comput. **27**(10), 1540–1554 (2009)
6. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychol. Rev. **94**, 115–147 (1987)
7. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3d mesh segmentation. ACM Trans. Graph. (SIGGRAPH) **28**(3) (2009)
8. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Meas. 37–46 (1960)
9. Corsini, M., Gelasca, E.D., Ebrahimi, T., Barni, M.: Watermarked 3d mesh quality assessment. IEEE Trans. Multim. **9**, 247–256 (2007)
10. Daniel, W.W.: A Foundation for Analysis in the Health Sciences Books, 7th edn. Wiley, New York (1999)
11. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Stat. Assoc. **78**(383), 553–569 (1983)
12. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. ACM Trans. Graph. (Proc. SIGGRAPH) (2004)
13. Gerig, G., Jomier, M., Chakos, A.: Valmet: A new validation tool for assessing and improving 3d object segmentation. In: MICCAI 2001: Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 516–523. Springer, Berlin (2001)
14. Giorgi, D., Biasotti, S., Paraboschi, L.: Shrec: shape retrieval contest: watertight models track. http://watertight.ge.imati.cnr.it/ (2007)
15. Golovinskiy, A., Funkhouser, T.: Randomized cuts for 3d mesh analysis. ACM Trans. Graph. **27**(5) (2008)
16. Ji, Z., Liu, L., Chen, Z., Wang, G.: Easy mesh cutting. Comput. Graph. Forum **25**(3), 283–291 (2006)
17. Katz, S., Tal, A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. Graph. (SIGGRAPH) **22**(3), 954–961 (2003)
18. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction. Vis. Comput. **21**(8–10), 649–658 (2005)
19. Lai, Y.K., Hu, S.M., Martin, R.R., Rosin, P.L.: Fast mesh segmentation using random walks. In: SPM '08: Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling (2008)
20. Lavoué, G., Dupont, F., Baskurt, A.: A new cad mesh segmentation method, based on curvature tensor analysis. Comput. Aided Des. **37**(10), 975–987 (2005)
21. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating algorithms and measuring ecological statistics. Int. Conf. Comput. Vis. **2**, 416–423 (2001)
22. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
23. Sander, P.V., Snyder, J., Gortler, S.J., Hoppe, H.: Texture mapping progressive meshes. In: SIGGRAPH 2001, pp. 409–416 (2001)
24. Shamir, A.: A survey on mesh segmentation techniques. Comput. Graph. Forum **27**(6), 1539–1556 (2008)
25. Shapira, L., Shamir, A., Cohen-Or, D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. Vis. Comput. **24**(4), 249–259 (2008)
26. Sheffer, A., Praun, E., Rose, K.: Mesh parameterization methods and their applications. Found. Trends Comput. Graph. Vis. (FTCGV) **2**(2), 64 (2007)
27. Tierny, J., Vandeborre, J.P., Daoudi, M.: Topology driven 3D mesh hierarchical segmentation. In: IEEE International Conference on Shape Modeling and Application (SMI) (2007)
28. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 929–944 (2007)
29. Zeckerberger, E., Tal, A., Shlafman, S.: Polyhedral surface decomposition with applications. Comput. Graph. **26**(5), 733–743 (2002)
30. Zhang, H., Fritts, J., Goldman, S.: Image segmentation evaluation: A survey of unsupervised methods. Comput. Vis. Image Underst. **110**, 260–280 (2008)

Halim Benhabiles received the engineer degree from University of Sciences and Technologies of Houari Boumediene (USTHB), Algiers, Algeria, in 2006, and the M.S. degree from the University of Sciences and Technologies of Lille (USTL), Lille, France, in 2008, both in computer science. He is currently a Ph.D. candidate within the Computer Science laboratory of the University of Lille (UMR USTL/CNRS 8022). He is also a teaching assistant in Telecom Lille1, School of Engineering. His research interests include shape modeling, shape similarity estimation, geometry processing and their applications.

Jean-Philippe Vandeborre received the M.Sc. degree in 1997 and the Ph.D. degree in Computer Science in 2002, both from Lille 1 University, France. Currently, he is an associate professor in Institut Telecom/Telecom Lille 1. He is also a member of the Computer Science Laboratory in Lille 1 University (LIFL UMR CNRS 8022). His research interests are mainly focused on three-dimensional model analysis, and include indexing and retrieval from content, 3D-mesh segmentation methods and automatic evaluation of segmentation quality. He is co-author of several papers in refereed journals and proceedings of international conferences. He has been involved in French and European projects and has served as program committee member and reviewer for international journals and conferences.

Guillaume Lavoué received the engineering degree in signal processing and computer science from CPE Lyon (2002), the MSc degree in image processing from the University Jean Monnet, St-Etienne (2002) and the PhD degree in computer science from the University Claude Bernard, Lyon, France (2005). After a postdoctoral fellowship at the Signal Processing Institute (EPFL) in Switzerland, since September 2006 he has been an associate professor at the French engineering university INSA of Lyon, in the LIRIS Laboratory (UMR 5205 CNRS). His research interests include 3D model analysis and processing, including compression, watermarking, perception, and 2D/3D recognition.



Mohamed Daoudi is a Full Professor of Computer Science in the Institut TELECOM; TELECOM Lille 1. He received the Ph.D. degree in Computer Engineering from the University of Lille 1 (USTL), France, in 1993 and Habilitation à Diriger des Recherches (HDR) from the University of Littoral, France, in 2000. He was the founder and the head of the MIIRE research group of LIFL (UMR USTL-CNRS 8022). His research interests include pattern recognition, image processing, invariant representation of images and shapes, three-dimensional analysis and retrieval and more recently 3D face recognition. He has published more than 80 papers in refereed journals and proceedings of international conferences. He is the author the book 3D processing: Compression, Indexing and Watermarking (Wiley, 2008). He has served as a Program Committee member for the International Conference on Pattern Recognition (ICPR) in 2004 and the International Conference on Multimedia and Expo (ICME) in 2004 and 2005. He is a frequent reviewer for IEEE Transactions on Pattern Analysis and Machine Intelligence and for Pattern Recognition Letters. His research has been funded by ANR, RNRT and European Commission grants. He is Senior Member of IEEE.

Annexe E

Learning boundary edges for 3d-mesh segmentation. Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. *Computer Graphics Forum - Eurographics Association*, 30(8) :2170–2182, December 2011.

Abstract. This paper presents a 3D-mesh segmentation algorithm based on a learning approach. A large database of manually segmented 3D-meshes is used to learn a boundary edge function. The function is learned using a classifier which automatically selects from a pool of geometric features the most relevant ones to detect candidate boundary edges. We propose a processing pipeline that produces smooth closed boundaries using this edge function. This pipeline successively selects a set of candidate boundary contours, closes them and optimizes them using a snake movement. Our algorithm was evaluated quantitatively using two different segmentation benchmarks and was shown to outperform most recent algorithms from the state-of-the-art.

DOI: 10.1111/j.1467-8659.2011.01967.x

COMPUTER GRAPHICS *forum*
Volume 0 (2011), number 0 pp. 1–13

Learning Boundary Edges for 3D-Mesh Segmentation

Halim Benhabiles[1], Guillaume Lavoué[2], Jean-Philippe Vandeborre[1,3] and Mohamed Daoudi[1,3]

[1]LIFL (UMR Lille1/CNRS 8022), University of Lille 1, France
{halim.benhabiles, jean-philippe.vandeborre, mohamed.daoudi}@lifl.fr
[2]Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
glavoue@liris.cnrs.fr
[3]Institut TELECOM, TELECOM Lille 1, France

Abstract

This paper presents a 3D-mesh segmentation algorithm based on a learning approach. A large database of manually segmented 3D-meshes is used to learn a boundary edge function. The function is learned using a classifier which automatically selects from a pool of geometric features the most relevant ones to detect candidate boundary edges. We propose a processing pipeline that produces smooth closed boundaries using this edge function. This pipeline successively selects a set of candidate boundary contours, closes them and optimizes them using a snake movement. Our algorithm was evaluated quantitatively using two different segmentation benchmarks and was shown to outperform most recent algorithms from the state-of-the-art.

Keywords: 3D-mesh, segmentation, learning, boundary edges, evaluation, ground-truth

ACM CCS: I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations—Boundary representations

1. Introduction

3D-mesh segmentation is a key ingredient for many applications such as indexing, compression, deformation, etc.

The recent creation of ground-truth databases for the segmentation of 3D-meshes [BVLD09, CGF09], has given to the computer graphics community the opportunity to quantitatively analyse and *learn* mesh segmentation. A very recent work based on learning has been proposed by Kalogerakis *et al.* [KHS10] and has demonstrated its efficiency through the improvement of the results over the state-of-the-art of mesh segmentation.

In this paper, we present a new fully automatic 3D-mesh segmentation algorithm based on boundary edge *learning*. Our algorithm is carried out using two main steps: an off-line step in which an objective boundary edge function is learned from a set of segmented training meshes, and an on-line step in which the learned function is used to segment any input 3D-mesh. The boundary function is learned using the AdaBoost classifier [FS97], which automatically selects from a set of geometric features the most relevant ones to detect candidate boundary edges. In the on-line step, the learned edge function is used successively to select a set of candidate boundary contours, to close them and to optimize them using a snake movement to produce the final segmentation. The best results are obtained when the learning is performed on objects from the same category as the object to segment. However, the results remain excellent even when we generalize the learning on different categories. Hence, we do not need to know the category of the input model to segment it.

Our algorithm provides very similar segmentations to those created by humans (Figure 1), and has been evaluated using the existing benchmarking systems [BVLD09, CGF09]. Our results outperform the best segmentation algorithms from the state-of-the-art in term of similarity to human segmentations. For instance, we achieved 8.8% Rand Index error on the Princeton segmentation benchmark [CGF09], whereas the last best result achieved on this database is 14.9% Rand Index error obtained by [GF08] (algorithm without learning), and 9.5% Rand Index error obtained by [KHS10]

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

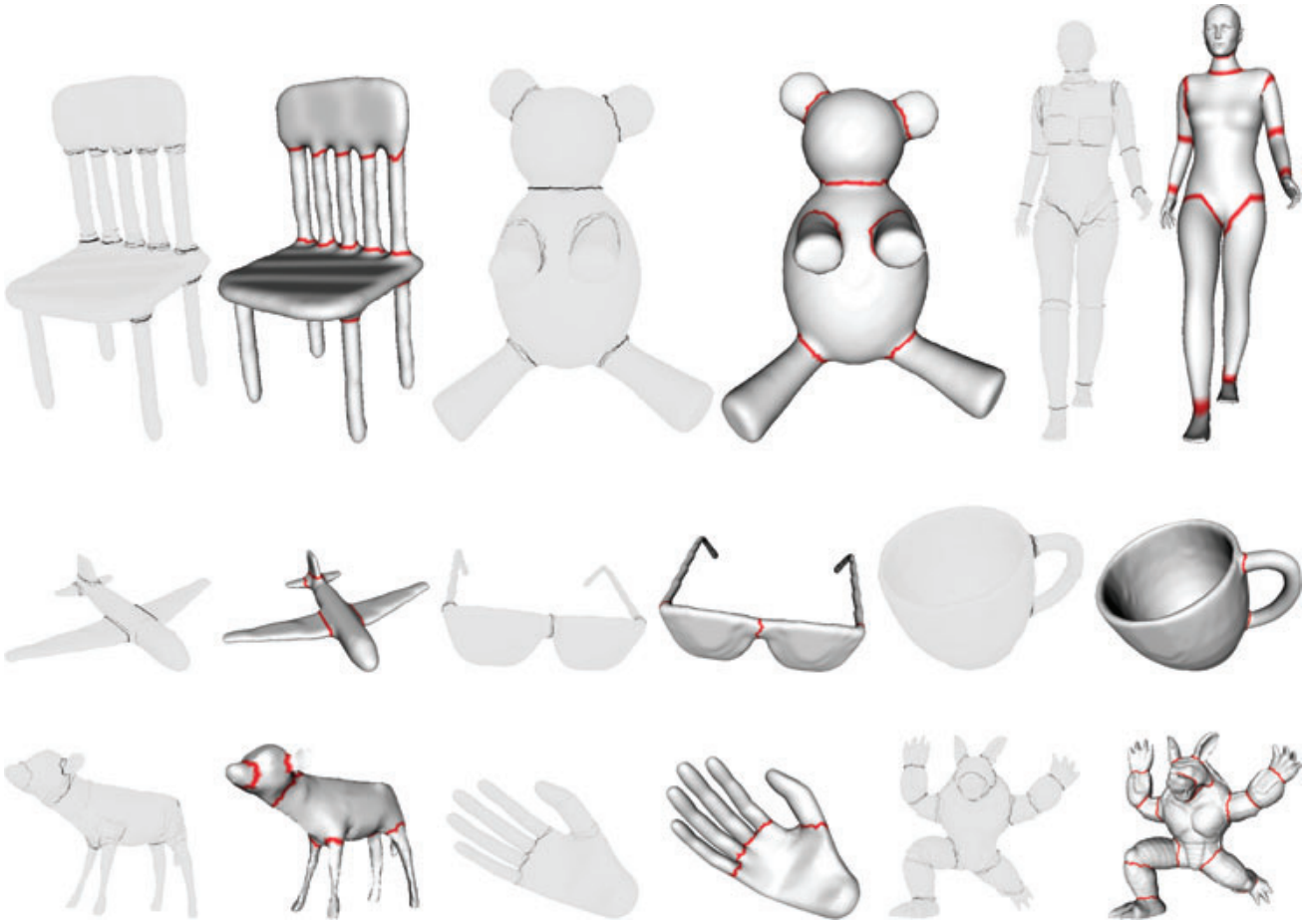


Figure 1: *For each pair of model: on the left, manual boundaries from the Princeton segmentation benchmark [CGF09] (the darkness degree of contours indicates that people have selected the same edges in their cuts); on the right, automatic boundaries from our algorithm.*

(algorithm with learning). This latter algorithm needs a consistent labelling of the training data, which may requires some manual interactions. However, it provides semantic labelling which allows for instance to retrieve a part of interest across a data-base of labelled 3D-models without additional processing.

Our main contributions are the following:

- The edge function determination based on multiple feature calculations and Adaboost learning.
- The processing pipeline which produces smooth closed boundaries using the edge function. This pipeline includes the following stages: thinning, closing contours, and a snake movement. More details are provided on each stage in Section 3.

2. Related Work

During the last decade, a significant attention has been paid, by the computer graphics community, to 3D-mesh segmentation [Sha08, AKM*06]. Although some supervised methods exist [LHMR08, JLCW06], most existing techniques are fully automatic. They either concentrate on the definition of contours and then extract regions limited by these contours [GF08, LLS*05], or use clustering of similar mesh elements to define regions [SNKS09, LW08, SSCO08]. In both cases, they focus on analysing either low-level geometric information, or topological information of the input mesh. For instance, the use of geometric criterion includes curvature [LDB05], geodesic distances [KT03], dihedral angles [ZTS02], shape diameter function [SSCO08], planarity and normal directions [AFS06], etc. The use of topological criterion includes mainly Reeb-graphs [TVD07] and spectral analysis [ZvKD07]. Such criteria suffer either from sensitivity to local surface features and to pose changes or from the deterioration of the topology when connecting or disconnecting parts of the mesh. As raised by Kalogerakis *et al.* [KHS10], the main drawback of this kind of algorithms is the fact that they are limited to a single generic rule (e.g. skeleton topology) or a single feature (e.g. curvature tensor). Indeed, such algorithm cannot be suited to segment an input 3D-mesh which requires a combination of these criteria.

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Our algorithm does not suffer from this drawback because it is based on a learning approach which determines an edge function using multiple criteria.

According to our knowledge, only one work has been proposed that involves learning for 3D-mesh segmentation [KHS10]. It allows to simultaneously segment and label the input mesh, and is expressed as an optimization problem. The problem consists in optimizing a conditional random field (CRF) of which an objective function is learned from a collection of labelled training meshes. We differ from this latter work in that instead of determining the suited label of each mesh facet and then implicitly defining a segmentation resulting from this labelling, we explicitly determine the boundary edges that allow then to obtain smooth closed contours that define the segmentation. Moreover, even complex boundaries can be captured (Section 3), while in the previous work, the method rather aims to find compact regions.

Before these recent works for 3D-mesh segmentation, several advanced works have already been introduced for 2D-image segmentation based on learning approaches. Like for the 3D case, these algorithms use a model learned from a database of segmented 2D-images. These 2D-image segmentation algorithms based on learning can be grouped into two categories.

The first category covers algorithms that learn an optimal affinity function between each pixel of the input image and a set of prior defined labels [SWRC09, HZCp04]. A ground-truth (segmented and labelled images) is employed to train the classifier that allows to affect the proper label to each pixel.

The second category covers algorithms that use an objective function to classify edges [MFM04, KH04]. Each edge is classified as a boundary or a non-boundary using a classifier trained on the ground truth (segmented images), resulting in an edge image estimating human designated boundaries.

Our algorithm is inspired by the second category since it classifies edges as boundary or not, while the previous 3D-work [KHS10] is inspired by the first category.

3. Our Segmentation Approach

In this section, we describe our approach. We provide details on the two main steps of our algorithm: the off-line step in which the objective boundary function is learned using a set of segmented models, and the on-line step in which the learned function is used to segment the input mesh.

3.1. Off-line (learning) step

We formulate the problem of learning the boundary edges as a classification problem. The classification model is learned on a corpus of 3D-meshes accompanied by their manual segmentations using the AdaBoost classifier. The classifier takes as input a training data set and generates a function. The training data set is composed of a set of feature vectors F_E computed for each edge of the ground-truth corpus meshes. A feature vector F_E of a given edge contains a set of geometric criteria and is associated with its proper class label L so that $L = +1$ if the edge is a boundary (according to the manual segmentations of the mesh containing this edge) and $L = -1$ if the edge is not a boundary. Once the learning is done, the classifier produces a function (the boundary edge function). This function takes as input a feature vector from any given edge and outputs a signed scalar value whose sign will provide the estimated classification of the edge (positive for boundary and negative for non-boundary).

Now we briefly summarize, the set of geometric criteria that we use to characterize edges (and which compose the feature vector), and the AdaBoost classifier.

3.1.1. *Feature vector*

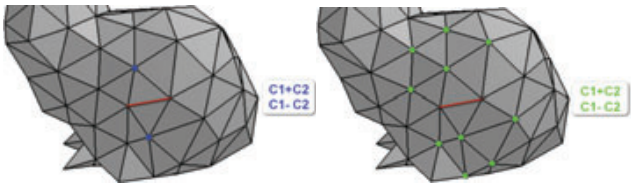
We compute a 33-dimensional feature vector for each edge. It includes a set of geometric criteria which are as follows:

- *Dihedral angle.* The angle between two adjacent facets.
- *Curvature.* We compute different curvatures using the VTK library (http://www.vtk.org/). Let K_1, K_2 be the principal curvatures, we include: K_1, K_2, $K_1 \times K_2$ (Gaussian curvature), $(K_1 + K_2)/2$ (mean curvature), $2/\pi \times \arctan[(K_1 + K_2)/(K_1 - K_2)]$ (shape index) and $\sqrt{(K_1^2 + K_2^2)/2}$ (curvedness).
- *Global curvature.* We compute the mean curvature integrated over a large geodesic radius (10% of the bounding box length) as in [LW08].
- *Shape diameter.* The shape diameter function [SSCO08] is computed using the default parameters: a cone with an opening angle of 120°, 30 rays, and a normalizing parameter $\alpha = 4$.

Note that we do not propose new geometric criteria, but we only employ existing ones used in previous segmentation algorithms. As stated in the introduction, the idea is to combine these criteria, then automatically select relevant ones with the appropriate weights using the classifier.

Except the dihedral angle which is computed for each edge, the other criteria (eight criteria) are computed for each vertex of the mesh. As illustrated in Figure 2 , to derive the criteria for each edge (the red one for instance), we consider its two opposite vertices (blue points in Figure 2a). Then, considering that C_1 and C_2 are the values of a certain criterion computed respectively on these two vertices, we derive two feature values for the edge: $C_1 + C_2$ and $C_1 - C_2$. The idea is that, according to the nature of the criterion, in certain cases the sum can be relevant while in others the difference can carry a better information.

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.



(a) (b)

Figure 2: *Example of edge criterion computation with one vertex on each side (a), and with a set of vertices (b).*

To bring a certain robustness to noise or sampling and to integrate a kind of multiresolution behaviour we also consider, in a second step, the one-level neighbourhood from each side of the edge (see green points in Figure 2b). In that case C_1 and C_2 are respectively the means of the criterion from vertices at each side of the edge. This yields 32 features (16 in each case) to which we add the angle feature.

3.1.2. *Adaboost classifier*

AdaBoost is a machine-learning method that builds a strong classifier by combining weaker ones [FS97]. The algorithm takes as input a set of training samples $(x_1, y_1), \ldots, (x_N, y_N)$; each x_i $(i = 1, \ldots, N)$ represents a feature vector (a vector F_E in our case), and each y_i represents the class label of the sample x_i (y_i belongs to the domain $Y = \{-1, +1\}$ in our case). A large number of weak classifiers, $h_i : X \longrightarrow Y$, are generated, each one being a decision stump associated with a single dimension of the feature vector. Then, along a finite number of iterations $(t = 1, 2, \ldots, T)$, the algorithm iteratively selects the weak classifier which minimizes the current classification error. At the end a strong classifier H is produced as the combination of the weak ones weighted with α_t: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$.

As stated at the beginning of Section 3.1, the generated function $H(x)$ is now able to produce a signed scalar for each edge of an input 3D-mesh, whose sign gives its class label (positive sign for boundary edges and negative sign otherwise).

Note that we tested some other existing classifiers from the literature including non-parametric and parametric models such as density estimation, hierarchical mixtures of experts (HME) and support vector machine (SVM). The performance was always nearly the same. We favour the AdaBoost because it yields a slight improvement over the other classifiers, and has the best running time.

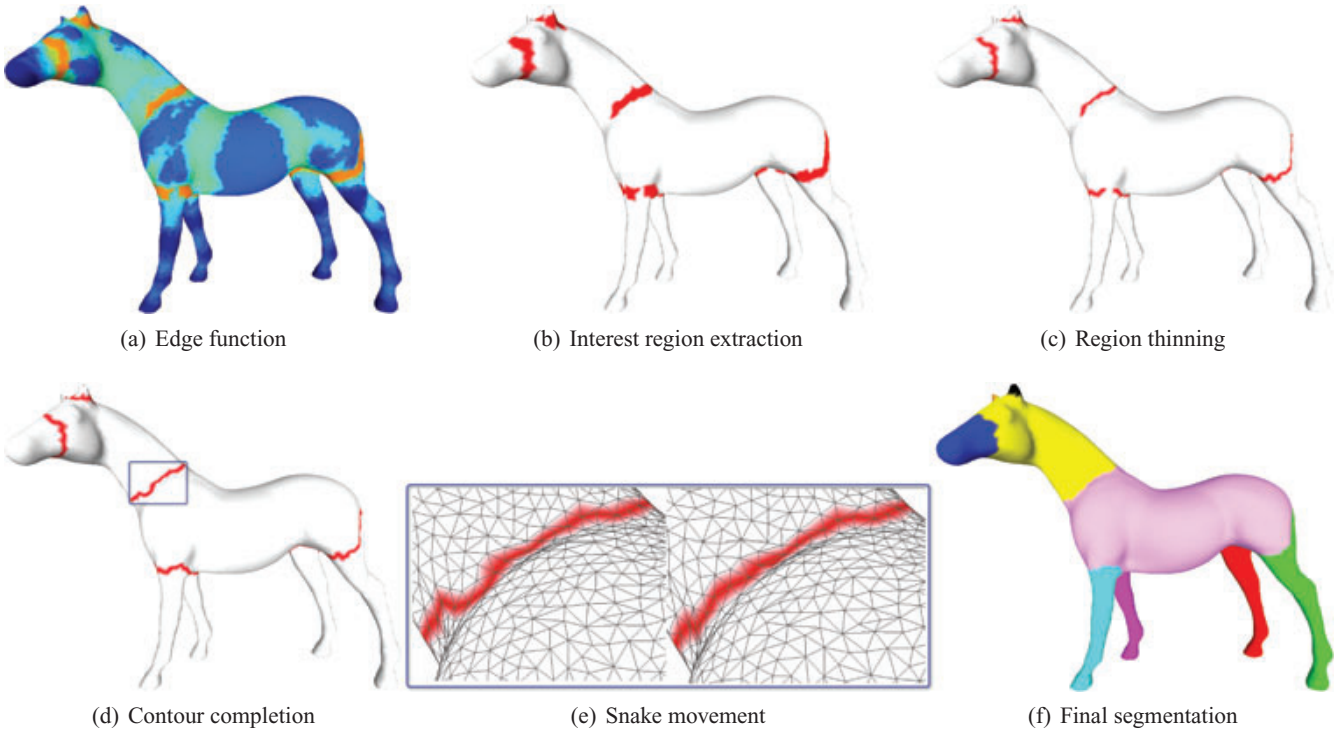
3.2. On-line (segmentation) step

Figure 3 shows examples of edge classification results of some 3D-meshes. On the top, the result of the binary decision: boundary $[H(x) > 0]$, non-boundary $[H(x) < 0]$ is displayed (for visualization quality reason, we coloured incident vertices of edges instead of colouring the edges themselves); whereas in the bottom, the edge function scalar values $H(x)$ are displayed using a colour map. One can notice that the boundary edges from the binary decision (in red) are, neither smooth, nor closed. This result is expected since our classification model is learned on different objects (even if they belong to the same category), and uses multiple ground-truths per model which are not necessarily the same (boundaries are defined in a subjective way by humans, see Figure 1). Hence it is not possible to directly consider this classifier output as a segmentation result.

Figure 3: *Edge classification results for some 3D-meshes (top: boundary edges after binary decision in red colour; bottom: edge function scalar field).*

To overcome this problem we propose a processing pipeline that transforms these non-connected fuzzy regions into thin, closed and smooth contours, by using the edge function. This processing pipeline comprises four stages. In the first stage of the process, given an input 3D-mesh, the edge function is computed (Figure 4a), and all edges having positive function values are selected. Theses edges constitute a set of interest regions (Figure 4b). Then, for each interest region (connected set of edges), a thinning algorithm [HG01]

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.



(a) Edge function (b) Interest region extraction (c) Region thinning

(d) Contour completion (e) Snake movement (f) Final segmentation

Figure 4: *Overview of the processing pipeline.*

is applied. This latter algorithm gives as output a set of open linear contours (Figure 4c). Next, each open contour is completed using an improved version of the algorithm proposed by [LLS*05] based on the edge function (Figure 4d). At this step we have created a set of closed contours which represent a first version of the segmentation boundaries. However, these boundaries are often not smooth nor precise since in the thinning stage we do not consider any geometric information. To overcome this drawback, we apply an improved version of the snake movement algorithm proposed by [JK04] based also on the learned edge function. The snake movement allows to improve the quality of the boundaries in term of smoothness and precision without changing the mesh connectivity (Figure 4e). This set of improved boundaries defines the final segmentation (Figure 4f). These steps are detailed in the following subsections. Material illustrating the different steps of the algorithm is available at the third author's home page.

Note that in all our experiments (more than 350 models), we never encountered any topological problem (e.g. broken regions representing the same boundary) like in the work from Lee *et al.* [LLS*05]. The main reason is probably that our regions of interest come from a learning processed on different models associated with different manual segmentations; hence it introduces a kind of fuzziness which smooth/filter the results. Thus, we create one closed boundary for each connected region, even if two regions are close to each other. We do not process any contour filtering or contour merging.

3.2.1. *Region thinning*

In this stage, we transform a set of interest regions into a set of thin contours; this set of contours will be further used as the initial set of boundaries. Each interest region is represented by a set of connected edges. The algorithm from Hubeli and Gross [HG01] allows to thin a given interest region to a piecewise linear contour by deleting the edges from the border of the patch toward the inside. Initially, the algorithm inserts all border edges into a list. A border edge is an edge of which at least one of the four edges of its two opposite triangles does not belong to the interest region. Then, each border edge is deleted from the interest region if it does not disconnect this latter one. The deleting operation produces new border edges which are added to the list. The algorithm is performed until there is no removable edge. This leads to produce a connected skeleton of the interest region. Moreover, the algorithm allows to obtain a directly closed contour if the interest region forms a loop. Note that this algorithm does not contain any parameter setting.

However, it is possible to obtain a branching skeleton. Figure 5 illustrates an example in which a model has an interest region that leads to the creation of a branching skeleton after undergoing a thinning algorithm. This skeleton is composed of external and internal branches. An external branch is limited by one endpoint and one junction point while an internal branch is limited by two junction points. We consider that only two external branches are correct regarding the real boundary and we consider others like noise; to

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

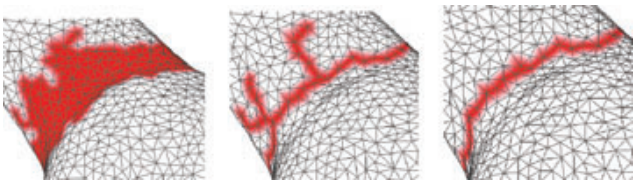


Figure 5: *From left to right: the interest region, the branching skeleton after thinning, the open boundary after removing the noisy branches for the horse model.*

select these two relevant branches we compute a weight for each external branch by summing the learned function values of their edges, and we keep the two branches that have the highest weights together with the internal branches that connect them (in the case where they do not share the same junction point). We precise here that according to our experiments, such branching skeletons appear mostly when the corresponding interest region is very large, which almost never happens.

3.2.2. *Contour completion*

In this stage, each open contour is completed to form a closed boundary around a specific part of the input mesh. To this end, we propose a modified version of the completion algorithm from Lee *et al.* [LLS*05]. The principle is to find the weighted shortest path between the two endpoints of the contour.

Let ζ be an open contour composed of a set of mesh vertices v. To close the open contour, we search the shortest path between the two end points of ζ by selecting among candidate edges using the following edge cost function:

$$cost(e) = \eta_d(e)^{w_d}.\eta_n(e)^{w_n}.\eta_e(e)^{w_e},$$

where $\eta_d(e)$ and $\eta_n(e)$ are defined as the average of values of the two incident vertices of the edge e.

η_d is the distance function. It measures the distance from ζ to a given vertex v as

$$\eta_d(v) = \sum_{v_i \in \zeta} \frac{1}{d(v, v_i)},$$

where $d(v, v_i)$ is the Euclidean distance. The function is high in the neighbourhood of the contour ζ and decreases otherwise.

η_n is the normal function. It helps to go over the other side of the mesh, and is defined for a given vertex v as

$$\eta_n(v) = \begin{cases} 1, & \text{if } n_\zeta.n_v \geq \cos(\alpha), \\ \dfrac{n_\zeta.n_v + 1}{\cos(\alpha) + 1}, & \text{else,} \end{cases}$$

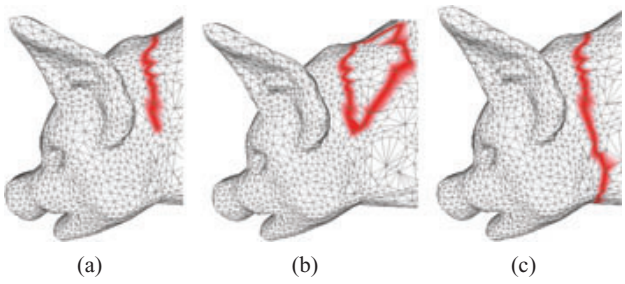


Figure 6: *Example of completing a contour on a 3D-mesh (a) using: the original version of the algorithm from Lee* et al. *[LLS*05] based on their feature function (b), and the improved version based on our learned edge function (c).*

where n_ζ is the average normal vector of ζ, n_v is the normal vector of vertex v and α is the angle between the normals of the two endpoints of ζ.

η_e is the feature function; in the original work from Lee *et al.* [LLS*05] the feature function includes minimum curvature and centricity. The centricity of a vertex is defined as the average geodesic distance from the given vertex to all other vertices of the mesh. As stated by the authors, the original algorithm sometimes failed to correctly close the open contours. In our modified version, we replace the feature function by our learned edge function; it guides the path towards the regions learned as boundaries according to the results of the classifier. The results are significantly improved (see an example in Figure 6).

Note that the three function values are normalized in the range [0, 1]. We set the weights w_d,w_n to 1 and w_e to 0.4.

3.2.3. *Snake movement*

The snake movement is used to optimize the set of closed contours resulting from the previous stage. Each contour is used as the initial position of the snake. We propose a modified version of the snake movement algorithm from Jung *et al.* [JK04]. The algorithm is based on an iterative process in which the snake evolves (each vertex of the snake is moved to one of its neighbour vertices on the mesh) by minimizing an energy functional composed of internal and external parts until it is adjusted (Figure 4e). In the original work from Jung *et al.* [JK04], the internal energy controls the length and the smoothness of the snake (i.e. the closed contour), and is defined for a given vertex v_i as

$$E_{\text{int}}(v_i) = \alpha \left\| v_i - v_{i-1} \right\| + \beta \left\| v_{i+1} - 2v_i + v_{i-1} \right\|,$$

where α and β are tuning parameters that affect respectively the smoothness of the snake in term of distance and curvature. We set the α to 0.2 and the β to 0.8. The external energy controls the fitting of the snake to the desired feature, and is defined in [JK04] by the maximum principal curvature.

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

In our modified version, we replace the maximum curvature by the learned edge function again. The external energy of a given vertex is then computed by averaging the edge function values of its incident edges, and normalizing them in the range [0, 1] after reversing the sign since we aim to minimize the energy. The modification of the external energy is justified by the fact that the original algorithm aims to find features related to ridges and valleys based only on a single geometric criterion (maximum curvature). Hence, when replacing the maximum curvature by the edge function, the quality of boundaries is clearly improved (see experiment 4.4), because this latter function is based on multiple geometric criteria which are learned to detect boundaries.

4. Experiments

Our segmentation method was evaluated and compared quantitatively to the most recent algorithms from the state-of-the-art. To this end, we used two recent benchmarks dedicated to 3D-mesh segmentation evaluation, namely the Princeton benchmark [CGF09] (http://segeval.cs.princeton.edu/), and the LIFL/LIRIS benchmark [BVLD09] (http://www-rech.telecom-lille1.eu/3dsegbenchmark/).

Note that we used the same control parameter values in all our experiments, except when we explicitly modify the threshold of the H function for the hierarchical segmentation experiment (Section 4.5). The different parameters are set as follows:

- H Function threshold: $H(x) > 0$.
- Thinning and branch-filtering: no parameter.
- Contour completion: w_d, w_n to 1 and w_e to 0.4 of $cost(e)$.
- Snake: α to 0.2 and β to 0.8 of E_{int}.

4.1. Segmentation results on the Princeton benchmark

The Princeton segmentation benchmark provides 19 categories of 3D-meshes, each one containing 20 3D-models accompanied with multiple ground-truth segmentations (manual segmentations). Our segmentation method was trained and tested on this benchmark, using different learning strategies, namely categorical learning and global learning.

In the first type of learning (categorical), we train and test our algorithm class by class. Similarly to [KHS10], we evaluate our method using leave-one-out cross-validation. For each mesh i in each class, we learn the edge function on the 19 other meshes from that class using all ground-truths, and then we use that function to segment the mesh i. To analyse the effect of the training set size on the quality of the results, we repeat the same experiment, using less meshes in the training set: we learned the edge function on six meshes randomly selected for each class.

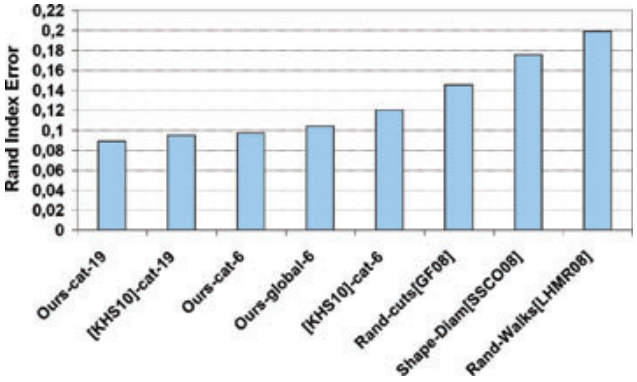


Figure 7: *Rand Index Error averaged over all the Princeton corpus models and sorted in increasing order for different algorithms. Reference-type-size represent the Index Error of algorithms based on learning with: learning type (categorical or global), size of the used training set (19 and 6 models).*

In the second type of learning (global), we learn the edge function in a generic way using a subset of the whole database (six models randomly selected from each category), then we test on the remaining models (14×19 models). In this generic (or global) learning scenario, we do not need to know the category of the model to segment.

To evaluate the quality of the segmentation produced by our algorithm, we follow the protocol defined in the Princeton segmentation benchmark. Figure 7 shows the Rand Index error averaged over all models of the corpus for our algorithm, using the different learning strategies, and for the most recent algorithms from the state-of-the-art. The first point to make is that, when using a categorical learning with a training set size of 19 models, our algorithm provides very high quality results; indeed, our algorithm yields the smallest Rand Index Error (8.8%) among all the other algorithms. One can notice also on the figure that when reducing the training set size to six models, and keeping a categorical learning, our algorithm still provides very good results with a slight drop of performance (9.7% Rand Index Error). The second point to make is that, our algorithm performs better than the algorithm from Kalogerakis *et al.* KHS10] which is also based on learning (with the same training set sizes: 19 and 6 models).

However, categorical learning involves the fact that before segmenting a model, you have to know its category hence it is not really fair to compare these results with generic algorithms [GF08, SSCO08, LHMR08, AFS06] which do not need such knowledge of the data. When using Global learning, our algorithm does not need this prior knowledge and thus can be compared with generic algorithms; the fact is that it significantly outperforms them: its Rand Index Error is 10.4% whereas the value of the second best [GF08] is 14%.

It is interesting to study which criteria are selected by the classifier in the learning step. Figure 8 illustrates the

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

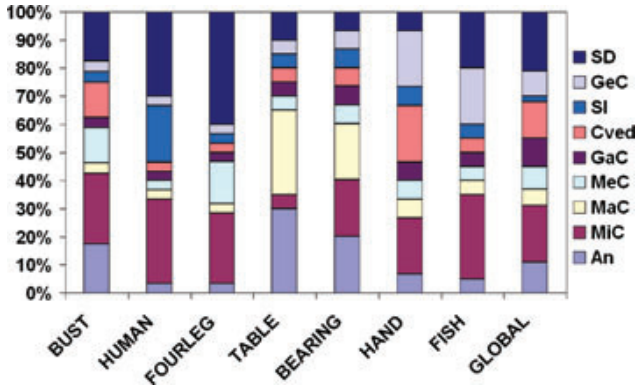


Figure 8: *Percentage of criteria selected by AdaBoost: for a categorical learning of size 19, and for a global learning of size 6. Legend: An (Angle), MiC (Minimum Curvature), MaC (Maximum Curvature), MeC (Mean Curvature), GaC (Gaussian Curvature), Cved (Curvedness), SI (Shape Index), GeC (Global Curvature), SD (Shape Diameter).*

percentage of each criterion selected by the AdaBoost classifier for both types of learning (categorical for several categories such as *bust*, *human*, etc. and global). We can notice that, whatever the learning strategy (categorical or global), all the criteria are selected by the classifier, hence they all contribute to the results. However, the distribution of the criteria percentages differs from a category to another. For instance, in the *fourleg* category, the most used criteria are the shape diameter and minimum curvature, while in the *table* category, the most used are the dihedral angle and maximum curvature. One can notice also a more isotropic distribution between the different criteria in the global learning with comparison to the categorical learning. This is due to the variety of 3D-meshes included in the training. All categories together may have different shapes and topologies, and then they do not share necessarily the same important features. In this case one or two criteria are clearly not sufficient to obtain a correct segmentation.

Figure 9 shows a visual comparison between our segmentation results (categorical learning—19 training models) and those from recent algorithms from the state-of-the-art on some 3D-meshes from the Princeton benchmark; the average of manual segmentations (ground-truths) is also included. The quality of our algorithm is confirmed; indeed, our segmentations appear better than those of the other methods in term of similarity to the ground-truths, and particularly regarding boundary precision.

4.2. Genericity of the learning across databases

In a second experiment, we still have trained our edge function on the Princeton benchmark but we have launched the segmentation on a different benchmark, the LIFL/LIRIS benchmark which contains a different set of 3D-models.

Figure 9: *From left to right segmentations obtained by: average of ground-truths of Princeton benchmark, our algorithm trained on the Princeton benchmark, [KHS10, GF08, SSCO08].*

Besides, the 3D-models are associated with vertex-based manual segmentations, while those of the Princeton benchmark are associated with facet-based segmentations. The LIFL/LIRIS benchmark contains 28 3D-models grouped in five classes. Each 3D-model is associated with four ground-truths. Similarly to the previous experiment, to evaluate the segmentation produced by our algorithm, we have followed the protocol defined in the benchmark. The edge function used to segment the models of this benchmark was learned on the Princeton segmentation benchmark using the global learning, with a training set size of six models. Figure 10 shows the normalized probabilistic rand index (NPRI) averaged over all the corpus models for different algorithms including ours. Contrary to the Rand Index Error, the NPRI [BVLD10] gives an indication about the similarity degree between the automatic segmentation and the manual segmentations. It is in the range $[-1,1]$. A value of 1 indicates a complete similarity, whereas a value of -1 indicates a maximum deviation between the segmentations being compared. The figure clearly shows the significant improvement of the

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

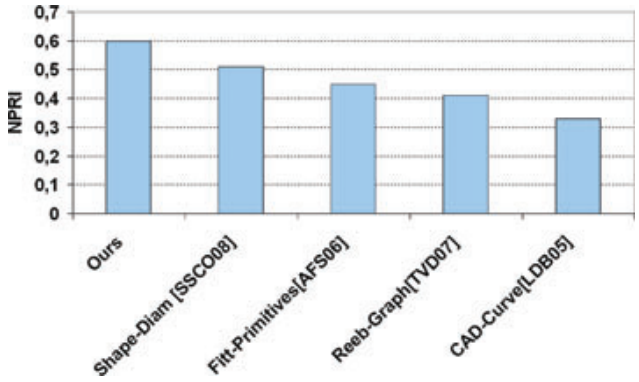


Figure 10: *Scores of NPRI (Normalized Probabilistic Rand Index) averaged over all the LIFL/LIRIS corpus models and sorted in decreasing order for different algorithms. Although in this experiment, our method is based on a global learning, performed on a different database, it outperforms the others.*

Figure 11: *Segmentations results, obtained by our algorithm trained on the Princeton benchmark, for a variety of meshes from different databases.*

results obtained by our method with comparison to the others. More precisely, our method reaches 60% of similarity rate, when the best result reached by the other methods on the same corpus is 50%. We have to precise here that these good results confirm the robustness and the genericity of our learning because we have trained our edge function on a different database containing different models.

Figure 11 shows segmentations obtained by our algorithm for some 3D-meshes selected from INRIA (http://www-roc.inria.fr/gamma/), TOSCA (http://tosca.cs.technion.ac.il/) and Stanford (http://graphics.stanford.edu/data/) databases. The edge function used to segment the models was learned on the Princeton segmentation benchmark (global learning, six models). Again, our algorithm correctly segments these meshes, and finds a set of meaningful parts.

4.3. Algorithm efficiency regarding the category of models

In this third experiment, we want to study the behaviour of our algorithm regarding the different categories of models in both benchmarks. For this reason we use the NPRI which is a more discriminative metric than a simple Rand Index (see [BVLD10] for more details). The NPRI is computed for each model, then averaged by category. Figure 12 illustrates the results obtained by our algorithm for the Princeton corpus models, and for the LIFL/LIRIS corpus models. Note that for the first corpus, we use a categorical learning of size 19, whereas for the second corpus, we use a global learning of size 6, both trained on the Princeton benchmark. Globally, we can notice that for both corpuses, our algorithm gives quite good results for each category since the scores are much higher than zero. An interesting point is that the scores of common categories among the two corpuses are consistent, with a slight drop of performance for the LIFL/LIRIS corpus which is due to the difference of learning strategy (categorical vs. global on a different database). The figure illustrates also that the *bust* category seems to be the most difficult one to segment (it is associated with the smallest NPRI values for both benchmarks). The fact is that human face images are well known in subjective experiments as a high-level factor attracting human attention, hence some features not relevant from a geometrical point of view can be considered highly relevant for human observers and thus can influence the manual ground-truth segmentations.

4.4. Study of the performance of our improved snake movement

In this experiment, we show how the use of the learned boundary edge function improves the original snake movement algorithm from Jung *et al.* [JK04]. To this end, we compute the similarity between boundaries extracted by our algorithm using both versions of the snake movement, and the manual boundaries on the LIFL/LIRIS corpus models. The most appropriate metric to compute this kind of similarity is the Cut Discrepancy Index (CDI) [BVLD10], because it allows to compute the mapping degree between the extracted boundaries of two segmentations of the same model. The metric is in the range $]-\infty; 1]$, and a value of 1 indicates a perfect matching between boundaries of the two segmentations. Figure 13 illustrates the scores of CDI averaged for each category of models and over all models of the LIFL/LIRIS corpus for these cases. The results show clearly that the new snake movement always improves the quality of the boundaries, whatever the category of models.

4.5. User interaction and coarse to fine segmentation

One of the strong point of our algorithm is that it is fully automatic; in particular, the number of boundaries (and thus

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

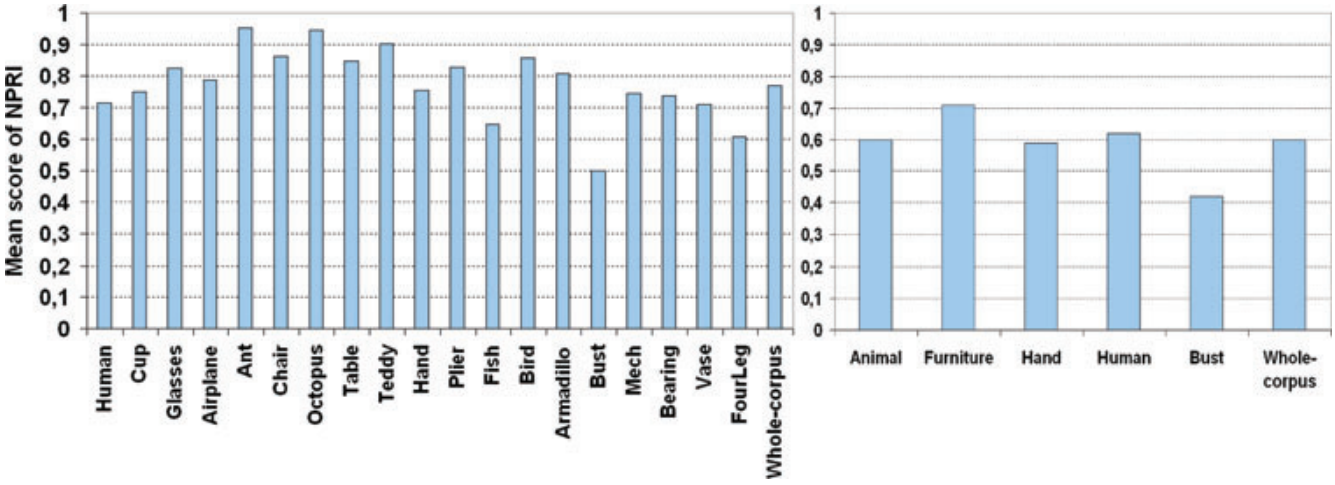


Figure 12: *From left to right, scores of NPRI averaged for each category and for all models from the Princeton benchmark [CGF09], and from the LIFL/LIRIS benchmark [BVLD09].*

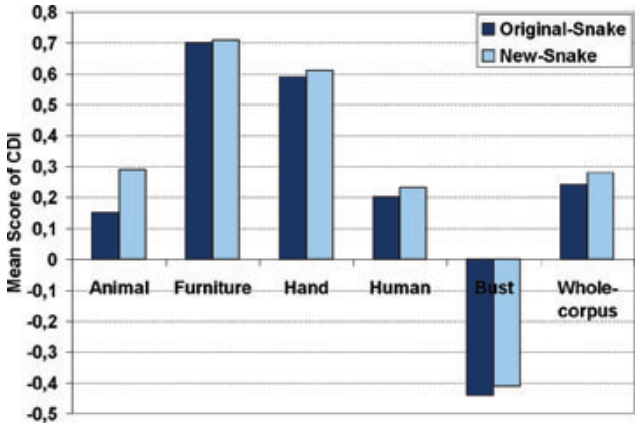


Figure 13: *Scores of CDI averaged for each category and over all models of the LIFL/LIRIS corpus with the original [JK04] and our improved version of the snake movement.*

(a) $H(x) > 0.3$ (b) $H(x) > 0$ (c) $H(x) > -0.2$

Figure 14: *Example of coarse to fine segmentation obtained by tuning the classification threshold applied on $H(x)$.*

the number of segments) is automatically determined within our processing pipeline; it corresponds to the number of connected interest regions from the edge classification step (Figure 4b). However if needed, it is quite easy to introduce human interactions in our process. This can be done by

- Tuning the classification threshold applied on the edge function. This threshold is set to 0 in our method; however, it is still possible to decrease (resp. increase) this threshold to obtain more (resp. less) segments. Such a coarse to fine segmentation is illustrated in Figure 14. We have to notice however that for a too low threshold (see the right hand in Figure 14) some regions of interest may become very large and thus may be abnormally merged leading to merging some segments (like the bottom of the fingers). However this kind of problems never happened in all our experiments, with a threshold set to 0.
- Using a paintbrush, to directly select on the mesh a set of edges representing a new interest region, similarly to [LLS*05]. The segmentation process is then completed by performing the remaining steps.

4.6. Algorithm robustness regarding geometric transformations

We assessed the robustness of our method against two kinds of transformations, namely pose-variation and noise.

- *Pose-variation:* Figure 15 shows the segmentations obtained by our algorithm for the armadillo and human

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Figure 15: *Algorithm robustness against pose-variation.*

models with different poses. These models are available in the Princeton segmentation benchmark. The edge function used to segment the models is based on a global learning, with a training set size of 6 × 19 models (six models selected from each category). Globally, the segmentation of the models is quite stable, which underlines the pose robustness of our algorithm.

- *Noise:* we applied on the vertices of the test models two random displacements in the direction of *x*-, *y*- and *z*-axis (3% and 5% of the bounding box length). Figure 16 shows the boundaries extracted by our algorithm for the noisy versions of the ant model. On the top, the boundaries are generated using an edge function based on a categorical learning, with a training set size of 19 models (all the training models belong to the ant category); whereas in the bottom, the edge function is based on a global learning, with a training set size of 6 × 19 (six models selected from each category). When applying a noise of 3%, the results remain very good for the categorical learning and correct for the global learning. However, when applying a strong noise (5%) the quality of the boundaries is seriously degraded particularly when using the global learning. We have conducted tests on 19 models (one model randomly selected from each category of the Princeton segmentation benchmark) for 3% and 5% of noise, the quality of segmentation decreases respectively of 1% and 12% when using the categorical learning, and of 10% and 50% when using the global learning. The quality is measured by computing the NPRI for each test model. This moderate robustness is due to the fact that the learning step is performed on clean data; hence the learned function fails to extract the right boundaries in the noisy models because the edges composing them do not share the same geometric properties as in the clean ones. A solution could be to artificially add noise on the segmented training data, before the feature extraction and learning.

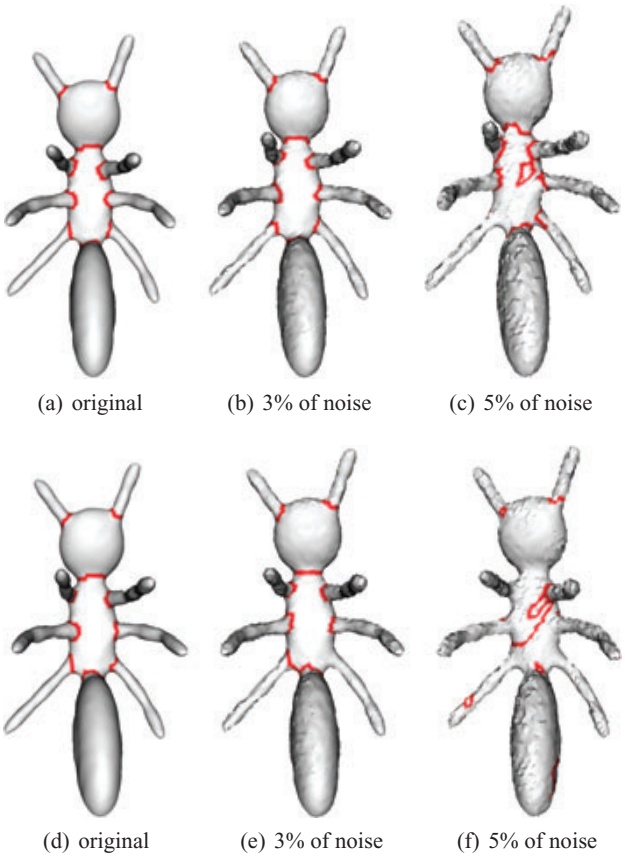


Figure 16: *Algorithm robustness against noise (top: boundary extraction based on categorical learning, bottom: boundary extraction based on global learning).*

4.7. Running time

All the previous experiments were carried out on a 2.99 GHz Intel(R) Core(TM) 2 Duo CPU with 3.25 Gb memory. Globally, the running time for the learning step is less than 10 min for most of the categories. The process is longer, around 40 min, for some categories of which the size of models is important such as the *armadillo* category. The online step (segmentation) runs at an interactive time (around 1 min), except for the *armadillo* category for which the running time is more important (around 9 min). This is due to the size of mesh and the number of extracted boundaries which are both important (24k vertices and 16 boundaries on average). More precisely, the average running time in seconds for the

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

thinning, contour completion, and snake movement steps is respectively 0.58, 3.81 and 45.95.

5. Conclusion

In this paper we have presented a framework for segmentation based on a learning approach. A boundary edge function is learned from a set of ground-truths using AdaBoost classifier, and then is used to segment any input 3D-mesh through different processing steps. Our framework is the first to address the problem of learning boundary edges for 3D-mesh segmentation. We have shown the possibility to make a generic (or global) learning that can lead to segment a model without having a prior information about its category. Our algorithm outperforms the most recent algorithms from the state-of-the-art, and produces segmentations which are very similar to those created by humans.

Although these results are accurate, some limitations remain and require a thorough analysis and future work. First, the number of geometric criteria used for classification could be more important. In particular, adding some rich features from 3D shape retrieval research fields would certainly improve the results and the sensitivity regarding to noise. The *bust* models are difficult to segment even with a training set of size 19, and our algorithm rather produces a coarse segmentation for this kind of models. This is probably due to the high semantic aspect carried by a face. This semantic aspect influences the manual segmentations, and is difficult to capture using simple local geometric criteria. Perhaps the solution is to include more global features such as topological ones; another solution would be to learn a whole structure of the segmentation per category (a prior graphical model from the manual segmentations), instead of a simple binary classification model.

Acknowledgments

This work is supported by the ANR (Agence Nationale de la Recherche, France) through MADRAS project (ANR-07-MDCO-015).

References

[AFS06] Attene M., Falcidieno B., Spagnuolo M.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer 22*, 3 (2006), 181–193.

[AKM*06] Attene M., Katz S., Mortara M., Patané G., Spagnuolo M., Tal A.: Mesh segmentation, a comparative study. In *IEEE International Conference on Shape Modeling and Applications (SMI)* (Matsushima, Japan, 2006).

[BVLD09] Benhabiles H., Vandeborre J.-P., Lavoué G., Daoudi M.: A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. In *IEEE International Conference on Shape Modeling and Applications (SMI)* (Beijing, China, 2009), pp. 36–43.

[BVLD10] Benhabiles H., Vandeborre J.-P., Lavoué G., Daoudi M.: A comparative study of existing metrics for 3D-mesh segmentation evaluation. *The Visual Computer 26*, 12 (2010), 1451–1466.

[CGF09] Chen X., Golovinskiy A., Funkhouser T.: A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics (SIGGRAPH) 28*, 3 (2009), 12. DOI: 10.1145/1576246.1531379.

[FS97] Freund Y., Schapire R. E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences 55*, 1 (1997), 119–139.

[GF08] Golovinskiy A., Funkhouser T.: Randomized cuts for 3d mesh analysis. *ACM Transactions on Graphics 27*, 5 (2008), 1–12.

[HG01] Hubeli A., Gross M.: Multiresolution feature extraction for unstructured meshes. In *Proceedings of the Conference on Visualization (VIS'01)* (San Diego, California, USA, 2001), pp. 287–294.

[HZCp04] He X., Zemel R. S., Carreira-perpiñán M. Á.: Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, 2004), pp. 695–702.

[JK04] Jung M., Kim H.: Snaking across 3d meshes. In *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications (PG'04)* (2004), IEEE Computer Society, pp. 87–93.

[JLCW06] Ji Z., Liu L., Chen Z., Wang G.: Easy mesh cutting. *Computer Graphics Forum 25*, 3 (2006), 283–291.

[KH04] Kaufhold J., Hoogs A.: Learning to segment images using region-based perceptual features. *IEEE Computer Vision and Pattern Recognition (CVPR) 2*, (2004), 954–961.

[KHS10] Kalogerakis E., Hertzmann A., Singh K.: Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics (SIGGRAPH) 29*, 3 (2010), 12. DOI: 10.1145/1833349.1778839.

[KT03] Katz S., Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (SIGGRAPH) 22*, 3 (2003), 954–961.

[LDB05] Lavoué G., Dupont F., Baskurt A.: A new cad mesh segmentation method based on curvature tensor

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

analysis. *Computer Aided Design 37*, 10 (2005), 975–987.

[LHMR08] LAI Y.-K., HU S.-M., MARTIN R. R., ROSIN P. L.: Fast mesh segmentation using random walks. In *Proceedings of the ACM Symposium on Solid and Physical Modeling* (Stony Brook, New York, USA, 2008), pp. 183–191.

[LLS*05] LEE Y., LEE S., SHAMIR A., COHEN-OR D., SEIDEL H.-P.: Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design 22*, 5 (2005), 444–465.

[LW08] LAVOUÉ G., WOLF C.: Markov random fields for improving 3d mesh analysis and segmentation. In *Proceedings of the Eurographics Workshop on 3D Object Retrieval* (Crete, Greece, 2008).

[MFM04] MARTIN D. R., FOWLKES C. C., MALIK J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26*, 5 (2004), 530–549.

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556.

[SNKS09] SIMARI P., NOWROUZEZAHRAI D., KALOGERAKIS E., SINGH K.: Multi-objective shape segmentation and labeling. *Computer Graphics Forum 28*, 5 (2009), 1415–1425.

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259.

[SWRC09] SHOTTON J., WINN J., ROTHER C., CRIMINISI A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision 81*, 1 (2009), 2–23. DOI: 10.1007/S11263_007_0109_1.

[TVD07] TIERNY J., VANDEBORRE J.-P., DAOUDI M.: Topology driven 3D mesh hierarchical segmentation. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI)* (Lyon, France, 2007).

[ZTS02] ZUCKERBERGER E., TAL A., SHLAFMAN S.: Polyhedral surface decomposition with applications. *Computers and Graphics 26*, 5 (2002), 733–743.

[ZvKD07] ZHANG H., VAN KAICK O., DYER R.: Spectral methods for mesh processing and analysis. In *Proceedings of the Eurographics State-of-the-Art Report* (Prague, Czech Republic, 2007), pp. 1–22.

© 2011 The Authors
Computer Graphics Forum © 2011 The Eurographics Association and Blackwell Publishing Ltd.

Annexe F

A new 3D-matching method of non-rigid and partially similar models using curve analysis. Hedi Tabia, Mohamed Daoudi, Jean-Philippe Vandeborre, and Olivier Colot. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4) :852–858, April 2011.

Abstract. The 3D-shape matching problem plays a crucial role in many applications such as indexing or modeling by example. Here, we present a novel approach to match 3D-objects in presence of non-rigid transformation and partially similar models. In this paper, we use the representation of surfaces by 3D-curves extracted around feature points. Indeed, surfaces are represented with a collection of closed curves and tools from shape analysis of curves are applied to analyze and to compare curves. The belief functions are used to define a global distance between 3D-objects. The experimental results obtained on the TOSCA and the SHREC07 datasets show that the system efficiently performs in retrieving similar 3D-models.

A New 3D-Matching Method of Nonrigid and Partially Similar Models Using Curve Analysis

Hedi Tabia,
Mohamed Daoudi, *Senior Member*, *IEEE*,
Jean-Philippe Vandeborre, *Member*,
IEEE, and Olivier Colot

Abstract—The 3D-shape matching problem plays a crucial role in many applications, such as indexing or modeling, by example. Here, we present a novel approach to matching 3D objects in the presence of nonrigid transformation and partially similar models. In this paper, we use the representation of surfaces by 3D curves extracted around feature points. Indeed, surfaces are represented with a collection of closed curves, and tools from shape analysis of curves are applied to analyze and to compare curves. The belief functions are used to define a global distance between 3D objects. The experimental results obtained on the TOSCA and the SHREC07 data sets show that the system performs efficiently in retrieving similar 3D models.

Index Terms—3D-shape matching, curves analysis, belief functions, feature points.

1 Introduction

For a few years, there has been increasing interest in analyzing the shapes of 3D objects. Advances in 3D scanning technologies, hardware-accelerated three-dimensional (3D) graphics, and related tools are enabling access to high-quality 3D data. As technologies are improving, the need for automated methods for analyzing the shapes of 3D objects is also growing. Shape matching remains a central technical problem for efficient 3D-object search engine design. It plays an important role in many applications, such as computer vision, shape recognition, shape retrieval, and computer graphics.

In recent years, a large number of papers have been concerned with 3D-shape matching. Most of the current 3D-shape matching approaches are based on global similarity. Some methods directly analyze the 3D mesh—such as Antini et al. [1] who use curvature correlograms—and some others use a 2D-view-based approach—such as Filali et al. [2] who propose an adaptive nearest neighbor-like framework to choose the characteristic views of a 3D model. These approaches enable retrieval of similar objects in spite of rigid transformations. Other approaches are also enabling retrieval of objects that differ from nonrigid transformations, such as shape bending or character articulation [3]. They are generally based on the graph or skeleton computation of a 3D object [4], [5].

Another case of shape matching is encountered when partially similar shapes are presented. This kind of shape matching plays a crucial role in many applications, such as indexing or modeling, by example. It is usually approached using the *recognition by part* idea [6], [7]: segmentation of the shape into significant parts, and matching pairs of parts as whole shapes. Bronstein et al. [8] also proposed considering the 3D-matching problem as a multicriterion optimization problem trying to simultaneously maximize the similarity and the significance of the matching parts.

In this paper, we present a new approach for 3D-object matching based on curve analysis. An important line of work in face analysis has been to use shapes of curves that lie on facial surfaces [9], [10]. However, these works have been intended for face recognition only and not for shape analysis of generic 3D surfaces. Our approach consists of representing 3D objects as a set of parts. Each part is represented by an indexed collection of closed curves in $\mathbb{R}^3$ on the 3D surface. Then, we use shape analysis of curves in the matching process. First of all, given a 3D object, a set of feature points is extracted. Around each feature point, a set of closed curves is automatically computed. One feature point and its associated set of curves is called a *3D part*. Curves are level curves of a surface distance function defined to be the length of the shortest path between that point and the feature point. This function is stable with respect to changes in nonrigid transformations. In this paper, we analyze the shape of 3D-part surfaces by analyzing the shape of their corresponding curves. The belief functions are used to define a global distance between 3D objects using the distance between their corresponding 3D parts.

The paper is organized as follows: A feature point extraction algorithm using geodesics is introduced in Section 2. Section 3 describes our framework for feature descriptors building, and explains how a set of curves is associated to each feature point already extracted. This section also presents a step-by-step procedure to compare 3D-part surfaces using the corresponding curves. In Section 4, we exploit the previously presented 3D-part surface comparison schema to propose a 3D-object matching method using fusion tools given by belief function theory. In Section 5, experimental results are shown. Finally, Section 6 contains the concluding remarks.

2 Feature Points Extraction

In this section, we present the algorithm used in the feature point extraction process. This algorithm is based on topological tools proposed by Tierny et al. [6]. Using a diversity of 3D surfaces, the proposed algorithm produces robust and well-localized feature points. The concept of feature points has been introduced by several authors [11], [12]. However, it is difficult to find a formal definition that characterizes this concept. In Katz et al. [12], feature points refer to the points localized at the extremity of a 3D surface's components. Formally, feature points are defined as the set of points that are the furthest away (in the geodesic sense) from all of the other points of the surface. They are equivalent to the local extrema of a geodesic distance function, which is defined on the 3D surface. Fig. 1 shows some 3D surfaces with their corresponding feature points.

Let v_1 and v_2 be the farthest vertices (in the geodesic sense) on a connected triangulated surface S. Let f_1 and f_2 be two scalar functions defined on each vertex v of the surface S as follows: $f_1(v) = \delta(v, v_1)$ and $f_2(v) = \delta(v, v_2)$, where $\delta(x, y)$ is the geodesic distance between points x and y on the surface.

As mentioned by [13], in a critical point classification, a local minimum of $f_i(v)$ is defined as a vertex v_{min} such that all of its level-one neighbors have a higher function value. Reciprocally, a local maximum is a vertex v_{max} such that all of its level-one neighbors have a lower function value. Let F_1 be the set of local extrema (minima and maxima) of f_1 and F_2 be the set of local extrema of f_2. We define the set of feature points F of the triangulated surface S as the closest *intersecting* points in the sets F_1 and F_2. In practice, f_1 and f_2 local extrema do not appear exactly on the same vertices, but in the same geodesic neighborhood.

- *H. Tabia and O. Colot are with LAGIS FRE CNRS 3303/University Lille 1, Cité Scientifique, Villeneuve d'Ascq Cedex, France. E-mail: hedi.tabia@telecom-lille1.eu, Olivier.Colot@univ-lille1.fr.*
- *M. Daoudi and J.-P. Vandeborre are with TELECOM Lille 1; Institut TELECOM/LIFL UMR CNRS 8022/University Lille 1, Cité Scientifique, Rue Guglielmo Marconi, BP 20145, 59653 Villeneuve d'Ascq, France. E-mail: {mohamed.daoudi, vandeborre}@telecom-lille1.eu.*

Manuscript received 13 July; revised 20 Jan. 2010; accepted 5 Aug. 2010; published online 8 Nov. 2010.
Recommended for acceptance by L. Van Gool.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-07-0448.
Digital Object Identifier no. 10.1109/TPAMI.2010.202.

0162-8828/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

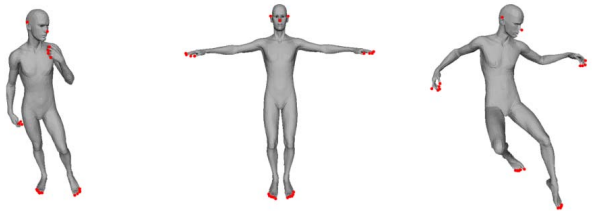


Fig. 1. Feature points extracted from different poses of a 3D model.

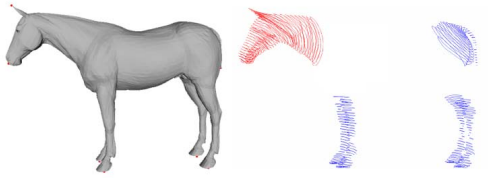


Fig. 2. 3D curves on the horse model extracted around its feature points.

Consequently, we define the intersection operator $\cap$ with the following set of constraints, where δ_n stands for the normalized geodesic distance function (to impose scale invariance):

$$V \in F = F_1 \cap F_2 \Leftrightarrow \begin{cases} \exists v_{F_1} \in F_1 \ / \ \delta_n(V, v_{F_1}) < \epsilon, \\ \exists v_{F_2} \in F_2 \ / \ \delta_n(V, v_{F_2}) < \epsilon, \\ \delta_n(V, v_i) > \epsilon \ \ \forall v_i \in F, \\ \epsilon, \delta_n \in [0, 1]. \end{cases}$$

This algorithm detects all of the feature points required in the subsequent analysis. They are accurately localized and their localization is robust with respect to rigid and nonrigid transformations because of the use of geodesic distance in f_1 and f_2 functions. In the experimental section, we will evaluate the localization accuracy of feature points detection within our global approach.

3 Feature Descriptors

As mentioned earlier, our goal is to analyze shapes of 3D surfaces using shapes of curves extracted around a set of feature points. In other words, we divide a 3D surface into a set of parts. Each part consists of one feature point and an indexed collection of simple, closed curves in $\mathbb{R}^3$. The geometry of 3D parts is then studied using the geometries of the associated curves. Curves are defined as level curves of an intrinsic distance function on the 3D-part surface. Their geometries, in turn, are invariant to the rigid transformations of the 3D-part surface. These curves jointly contain all of the information about the surface and it is possible to go back and forth between the surface and the curves without any ambiguity. In this section, we first describe how to extract a significant set of curves around feature points. We then explain the framework of curves analysis and its extension to comparison of 3D-part surfaces. Let V_i be a feature point on a 3D-triangulated surface. A geodesic distance function is defined on that 3D surface such that V_i is its origin. The geodesic function is split into a set of levels. Vertices which are in the same level are extracted with respect to an arbitrary order. Let λ be a level set corresponding to the geodesic distance function f. The set of ordered vertices v such that $f(v) = \lambda$ builds one curve. Fig. 2 shows the sets of curves corresponding to the horse's feature points.

3.1 Curves Analysis

The analysis of 3D shapes via their associated curve space has a technical sound foundation in Morse theory [14]. In our approach, we treat curves as closed, parameterized in $\mathbb{R}^3$, and we rescale them to have the same length, say 2π. This allows us to use one of many methods already available for elastic analysis of closed curves.

Several authors, starting with Laurent [15], followed by Michor and Mumford [16] and others, have studied curves for planar shapes. More recently, Joshi et al. [17] have extended it to curves in $\mathbb{R}^n$ using an efficient representation of curves. Other authors, including Yezzi and Mennucci [18], have also used Riemannian metrics on curve spaces. Their main purpose was to study curves evolution rather than shape analysis. Here, we adopt Joshi et al.'s approach [17] because it simplifies the elastic shape analysis. The main steps are: 1) Define a space of closed curves of interest, 2) impose a Riemannian structure on this space using the elastic metric, and 3) compute geodesic paths under this metric. These geodesic paths can then be interpreted as optimal elastic deformations of curves.

We start by considering a closed curve β in $\mathbb{R}^3$. Since it is a closed curve, it is parameterizable using $\beta : \mathbb{S}^1 \to \mathbb{R}^3$. We will assume that the parameterization is nonsingular, i.e., $\|\dot{\beta}(t)\| \neq 0$ for all t. The norm used here is the euclidean norm in $\mathbb{R}^3$. Note that the parameterization is not assumed to be arc length; we allow a larger class of parameterizations for improved analysis. To analyze the shape of β, we shall represent it mathematically using a *square-root velocity function* (SRVF), denoted by $q(t)$, according to which:

$$q(t) \doteq \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}}.$$

The $q(t)$ is a special function that captures the shape of β and is particularly convenient for shape analysis, as we describe next. First, the squared $\mathbb{L}^2$-norm of q, given by: $\|q\|^2 = \int_{\mathbb{S}^1} \langle q(t), q(t) \rangle dt = \int_{\mathbb{S}^1} \|\dot{\beta}(t)\| dt$, which is the length of β. Therefore, the $\mathbb{L}^2$-norm is convenient to analyze curves of specific lengths. Second, as shown in [17], the classical elastic metric for comparing shapes of curves becomes the $\mathbb{L}^2$-metric under the SRVF representation. This point is very important as it simplifies the calculus of the elastic metric to the well-known calculus of functional analysis under the $\mathbb{L}^2$-metric. In order to restrict our shape analysis to closed curves, we define the set: $\mathcal{C} = \{q : \mathbb{S}^1 \to \mathbb{R}^3 | \int_{\mathbb{S}^1} q(t)\|q(t)\| dt = 0\} \subset \mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^3)$.

Here, $\mathbb{L}^2(\mathbb{S}^1, \mathbb{R}^3)$ denotes the set of all functions from $\mathbb{S}^1$ to $\mathbb{R}^3$ that are square integrable. The quantity $\int_{\mathbb{S}^1} q(t)\|q(t)\| dt$ denotes the total displacement in $\mathbb{R}^3$ as one traverses along the curve from start to end. Setting it equal to zero is equivalent to having a closed curve. Therefore, $\mathcal{C}$ is the set of all closed curves in $\mathbb{R}^3$, each represented by its SRVF. Notice that the elements of $\mathcal{C}$ are allowed to have different lengths. Due to a nonlinear (closure) constraint on its elements, $\mathcal{C}$ is a nonlinear manifold. We can make it a Riemannian manifold by using the metric; for any $u, v \in T_q(\mathcal{C})$, we define:

$$\langle u, v \rangle = \int_{\mathbb{S}^1} \langle u(t), v(t) \rangle dt. \tag{1}$$

We have used the same notation for the Riemannian metric on $\mathcal{C}$ and the euclidean metric in $\mathbb{R}^3$, hoping that the difference is made clear by the context. For instance, the metric on the left side is in $\mathcal{C}$, while the metric inside the integral on the right side is in $\mathbb{R}^3$. For any $q \in \mathcal{C}$, the tangent space: $T_q(\mathcal{C}) = \{v : \mathbb{S}^1 \to \mathbb{R}^3 | \langle v, w \rangle = 0, w \in N_q(\mathcal{C})\}$, where $N_q(\mathcal{C})$, the space of normals at q, is given by:

$$N_q(\mathcal{C}) = \text{span}\Big\{ \frac{q^1(t)}{\|q(t)\|} q(t) + \|q(t)\| \mathbf{e}^1, \frac{q^2(t)}{\|q(t)\|} q(t) + \|q(t)\| \mathbf{e}^2, \frac{q^3(t)}{\|q(t)\|} q(t) + \|q(t)\| \mathbf{e}^3 \Big\},$$

and where $\{\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3\}$ form an orthonormal basis of $\mathbb{R}^3$.

It is easy to see that several elements of $\mathcal{C}$ can represent curves with the same shape. For example, if we rotate a curve in $\mathbb{R}^3$, we get a different SRVF but its shape remains unchanged. Another similar situation arises when a curve is reparameterized; a

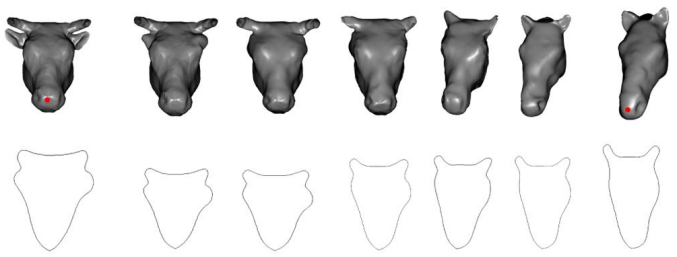


Fig. 3. Geodesic path between the cow-head part and the horse-head part.

reparameterization changes the SRVF of curve but not its shape. In order to handle this variability, we define orbits of the rotation group $SO(3)$ and the reparameterization group Γ as the equivalence classes in $\mathcal{C}$. Here, Γ is the set of all orientation-preserving diffeomorphisms of $\mathbb{S}^1$ (to itself) and the elements of Γ are viewed as reparameterization functions. For example, for a curve $\beta : \mathbb{S}^1 \to \mathbb{R}^3$ and a function $\gamma : \mathbb{S}^1 \to \mathbb{S}^1$, $\gamma \in \Gamma$, the curve $\beta(\gamma)$ is a reparameterization of β. The corresponding SRVF changes according to $q(t) \mapsto \sqrt{\dot{\gamma}(t)}q(\gamma(t))$. We set the elements of the set: $[q] = \{\sqrt{\dot{\gamma}(t)}Oq(\gamma(t))|O \in SO(3), \gamma \in \Gamma\}$ to be equivalent from the perspective of shape analysis. The set of such equivalence classes, denoted by $\mathcal{S} \doteq \mathcal{C}/(SO(3) \times \Gamma)$, is called the *shape space* of closed curves in $\mathbb{R}^3$. Here, $\mathcal{S}$ inherits a Riemannian metric from the larger space $\mathcal{C}$ and is thus a Riemannian manifold itself. The main ingredient in comparing and analyzing shapes of curves is the construction of a geodesic between any two elements of $\mathcal{S}$, under the Riemannian metric given in (1). Given any two curves β_1 and β_2, represented by their SVRFs q_1 and q_2, we want to compute a geodesic path between the orbits $[q_1]$ and $[q_2]$ in the shape space $\mathcal{S}$. This task is accomplished using a *path-straightening approach*, which was introduced in [19]. The basic idea here is to connect the two points $[q_1]$ and $[q_2]$ by an arbitrary initial path α and to iteratively update this path using the negative gradient of an energy function $E[\alpha] = \frac{1}{2}\int_s \langle \dot{\alpha}(s), \dot{\alpha}(s)\rangle ds$. The interesting part is that the gradient of E has been derived analytically and can be used directly for updating α. As shown in [19], the critical points of E are actually geodesic paths in $\mathcal{S}$. Thus, this gradient-based update leads to a feature point of E which, in turn, is a geodesic path between the given points. We will use the notation $d(\beta_1, \beta_2)$ to denote the geodesic distance, or the length of the geodesic in $\mathcal{S}$, between the two curves β_1 and β_2. Shown in the bottom row of Fig. 3 is an example of geodesic path between two 3D curves extracted from two different 3D-part surfaces: the cow-head part (left side) and the horse-head part (right side).

3.2 3D-Parts Matching

Now we extend ideas developed in the previous section for analyzing shapes of curves to the shapes of full 3D-part surfaces. As mentioned earlier, a 3D-part surface P is represented with an indexed collection of the level curves of the f function. That is, P is equivalent to the set $\{c_\lambda \in [0, L]\}$, where c_λ is the level set associated with the distance function value equal to λ. Through this relation, each 3D part has been represented as an element of the set $\mathcal{C}^{[0,L]}$. In our framework, the shapes of any two parts are compared by comparing their corresponding curves. Let D denote the global distance between two parts.

Given any two surfaces parts P_1 and P_2 and their collection of curves $\{c^1_\lambda, \lambda \in [0, L]\}$ and $\{c^2_\lambda, \lambda \in [0, L]\}$, respectively, our idea is to compare the curves c^1_λ and c^2_λ, and to accumulate these distances over all λ. Formally, we define a distance $D : \mathcal{C}^{[0,L]} \times \mathcal{C}^{[0,L]} \to \mathbb{R}_{\geq 0}$, given by

$$D(P_1, P_2) = \int_0^L d(c^1_\lambda, c^2_\lambda) d\lambda.$$

Here, the distance inside the integral is the geodesic distance function between the shapes of any curves, described in the last section. In Fig. 3, the top row shows the two 3D-part surfaces (cow head and horse head) and the resulting geodesic path between them. The middle surfaces denote five equally spaced points (in the space of all parameterized paths) along the geodesic path. With respect to the chosen Riemannian metric, this path denotes the optimal deformation from the cow-head part to the horse-head part.

4 3D-Shape Matching using Belief Function

In this section, we use the proposed 3D-parts matching method in order to compute a global distance between 3D objects. For that, a fusion method based on the *Transferable Belief Model* (TBM) framework [20], an interpretation of the Dempster-Shafer theory of evidence, is proposed and applied to perform the 3D-shape matching. The TBM is shown to provide a powerful framework, well suited to the merge process.

4.1 Transferable Belief Model Concept

The TBM is based on a two-level model: a credal level where beliefs are entertained, combined, and updated, and a pignistic level where beliefs are converted into probabilities to make decisions.

4.1.1 Credal Level

Let Ω denote a finite set called the frame of discernment. A *Basic Belief Assignment* (BBA) or mass function is a function $m : 2^\Omega \to [0, 1]$, such that: $\sum_{A \subseteq \Omega} m(A) = 1$, $m(A)$ measures the amount of belief that is committed to A. The subsets A of Ω such that $m(A) > 0$ are called focal elements.

Given two BBAs m_1 and m_2 defined over the same frame of discernment Ω and induced by two distinct pieces of information, we can combine them using the Dempster-Shafer Combination rule [21] given by:

$$(m_1 \otimes m_2)(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \tag{2}$$

for all $A \subseteq \Omega$.

4.1.2 Pignistic Level

When a decision has to be made, the beliefs held at the credal level induce a probability measure at the pignistic level. Hence, a transformation from belief functions to probability functions must be done. This transformation is called the pignistic transformation. Let m be a BBA defined on Ω; the probability function induced by m at the pignistic level, denoted by $BetP$ and also defined on Ω is given by:

$$BetP(\omega) = \sum_{\omega \in A} \frac{m(A)}{|A|}, \tag{3}$$

for all $\omega \in \Omega$ and where $|A|$ is the number of elements of Ω in A.

4.2 3D-Shape Matching

Given a collection of M 3D objects $\Omega = \{O_j, 1 \leq j \leq M\}$ and a set of N 3D-object parts $\{P_i, 1 \leq i \leq N\}$ extracted from a global 3D-object query Q, our aim is to compute a global similarity metric between the 3D-object query Q and 3D objects in the collection. This metric is based on the distance between parts presented in Section 3. In order to achieve this computation, each part P_i is modeled by one BBA m_{P_i}. This BBA is defined on Ω as frame of discernment and it measures the amount of belief accorded to the assumption: P_i belongs to O_j. Formally, m_{P_i} is a mass function $2^\Omega \to [0, 1]$ given by:

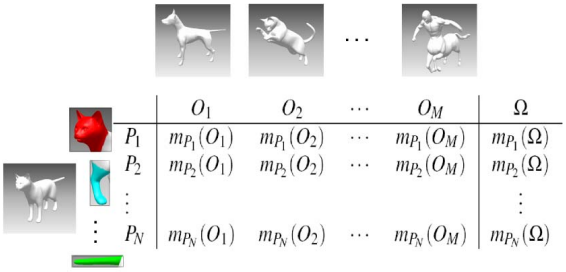


Fig. 4. Cat parts and their respective BBAs.

$$m_{P_i}(O_j) = \mu(P_i) \cdot \frac{1 - D_1(P_i, O_j)}{\sum_{l=1}^{M}(1 - D_1(P_i, O_l))}, \qquad (4)$$
$$m_{P_i}(\Omega) = 1 - \mu(P_i),$$

where D_1 is a distance measure defined between an object part P_i of the query Q and a global 3D object O_j in the collection. This measure represents the distance between the part P_i and its closest one in O_j. Formally, D_1 is given by:

$$D_1(P_i, O_j) = \min_{k, P_k \subset O_j} (D(P, P_k)). \qquad (5)$$

It is required to normalize this distance in order to obtain a distribution on $[0, 1]$, which allows us a correct construction of the BBA.

In (4), $\mu(P_i)$ is a confidence coefficient on $[0, 1]$. This coefficient is interpreted as the significance given to each part in a global 3D object.

As mentioned by Bronstein et al. [8], the quantification of the significance of an object's part is a major challenge. Here, we define the significance of a shape's part as a function of the partiality function used in [8]. The value $\mu(P_i)$ of a 3D part can be computed as $\mu(P_i) = exp(-npartiality(P_i)^2)$, where $npartiality$ is the normalized partiality function of the 3D part P_i. It is given by $npartiality(P_i) = \frac{area(Q) - area(P_i)}{area(Q)}$.

When all parts P_i of the query Q are modeled by their corresponding mass functions, we can apply the Dempster-Shafer combination rule in order to get a mass function which measures the amount of belief committed to the assumption: Q is similar to O_j.

Last, a decision is made from the obtained BBA and resulting objects are sorted based on a pignistic probability distribution. Our approach is summarized in Algorithm 1 and depicted in Fig. 4. In Fig. 4, a 3D object of a cat is divided into several parts P_i. A mass distribution BBA is constructed for each part over all objects in the collection. The combination of all masses gives us a similarity indication about the whole object with all objects in the collection.

Algorithm 1. 3D-shape search algorithm
Given a 3D-object query Q.
1: *Extract a set of N feature points V_i from Q.*
2: **for** *each feature point V_i where $i = 1, \ldots, N$* **do**
3: *Extract a set of closed curves (i.e., part P_i).*
4: *Compute the distance between P_i and all objects in Ω using (5).*
5: *Set a BBA value for all the objects $m_{P_i}(O_j)$ using (4).*
6: **end for**
7: *Combine all BBAs m_{P_i} into a new BBA m_Q using (2).*
8: *Compute the pignistic probability induced by m_Q using (3).*
9: *Display 3D objects in the order according to the pignistic probability.*

5 Experiments and Results

In this section, we present the databases, some state-of-the-art shape-matching algorithms to compare with, the evaluation criteria, and the experimental results. Data with noise are considered in the sequel, and a quantitative analysis of the experimental results is reported. The algorithms that we have described in the previous sections have been implemented using MATLAB software. The framework encloses an offline feature point extraction algorithm, a curve extraction algorithm, and an online retrieval process. To evaluate our method, we used two different databases. The TOSCA data set for nonrigid shapes and the SHREC07 shape benchmark for partially similar models.

Fig. 5. SHREC07 query-set snapshot.

5.1 TOSCA Data Set Description

The TOSCA[1] data set has been proposed by Bronstein et al. [22]. It is an interesting database for nonrigid shape correspondence measures. The database consists of 148 models, enclosing 12 classes. Each class contains one 3D shape under a variety of poses (between 1 and 20 poses for each class). This classification is provided with the data set.

5.2 SHREC07 Data Set Description

This benchmark[2] is composed of a data set of 400 manifold models and of a query set of 30 manifold models composed of composite models as shown in Fig. 5. Hence, it is an interesting database for partial shape retrieval. The data set exhibits diverse variations, from pose change to shape variability, within a same class or topology variation (notice 4 of the 20 classes contain nonzero genus surfaces) [23]. The ground truth is provided with the data set.

5.3 Some State-of-the-Art Algorithms

In order to evaluate the 3D-global matching approach, we compare our method with some state-of-the-art shape-matching algorithms.

- **Extended Reeb Graphs** (ERG): This is a structural-based 3D-matching method. It works with Reeb graph properties [23].
- **Ray-Based Approach with Spherical Harmonic Representation** (RSH): Saupe and Vranic [24] use PCA to align the models into the canonical position. Then the maximal extents are extracted, and finally, the spherical harmonic is applied.
- **The Hybrid Feature Vector** (DSR): This is a combination of two view-based descriptors: the depth buffer and the silhouette and radialized extent function descriptor [25].
- **The Geodesic D2**: This is an extension of the euclidean D2 [26]. It is computed as a global distribution of geodesic distances in 3D shapes.

5.4 Evaluation Criterion

There are several different performance measures which can evaluate retrieval methods. In this paper, we test the robustness of our approach using the Precision versus Recall plots. They are well known in the literature of content-based search and retrieval. The precision and recall are defined as follows:

$$Precision = \frac{N}{A}, \quad Recall = \frac{N}{Q},$$

where N is the number of relevant models retrieved in the top A retrievals, Q is the number of relevant models in the collection, which is the number of models to which the query belongs to.

1. http://tosca.cs.technion.ac.il/.
2. http://partial.ge.imati.cnr.it/.

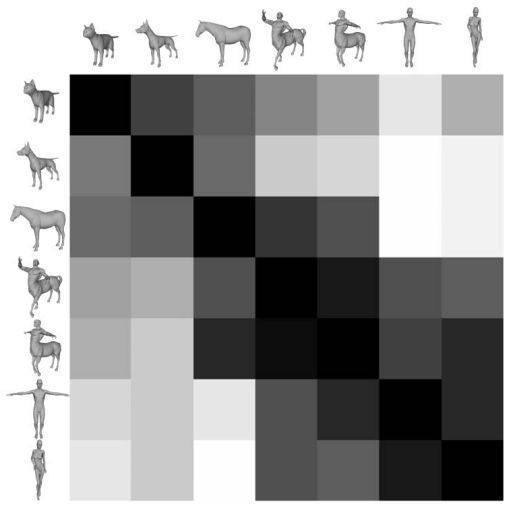


Fig. 6. Matrix of pairwise distances between seven 3D objects.

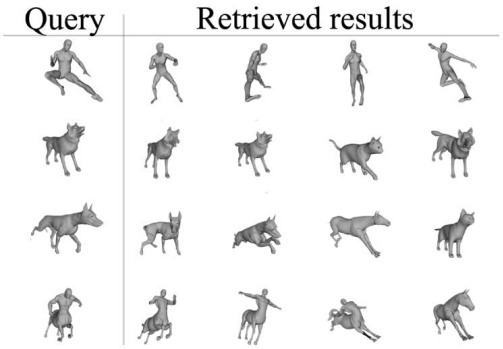


Fig. 7. First retrieved results on the TOSCA data set.

5.5 Results

During the offline process, each model in the database has been remeshed to get a regular model. Feature points have been extracted and their related sets of curves (parts) have been extracted and stored into indexed files. During the online process, feature points and related curves of the query object are extracted.

5.5.1 Results on the TOSCA Data Set

In order to show the main contribution of our approach, some results are shown as a matrix in Fig. 6. In this visualization of the matrix, the lightness of each element $(i; j)$ is proportional to the magnitude of the distances between 3D objects i and j. That is, each square in this matrix represents the distances between two 3D objects. Darker elements represent better matches, while lighter elements indicate worse matches.

This matrix is not symmetric because our similarity measure based on belief function is not symmetric either. In this matrix, humans are similar together and partially similar to centaurs. Centaurs are similar together and partially similar to horses and humans. According to the lightness of the matrix square, we can easily make the distinction among three different classes. The first class contains the first three animals, the second class encloses centaurs and horse, and the third class contains humans and centaurs.

Other visual results are shown in Fig. 7. The first column contains the query, while the rest contain the four retrieved 3D objects. An interesting result is shown in the last row of Fig. 7, where the query is a centaur; the fourth retrieved 3D object is a horse, which is partially similar to the centaur.

Quantitative analysis of our experimental results is depicted in Fig. 8. Fig. 8 shows the Precision versus Recall plots for our approach and some well-known descriptors. We find that our approach provides the best retrieval precision in this experiment.

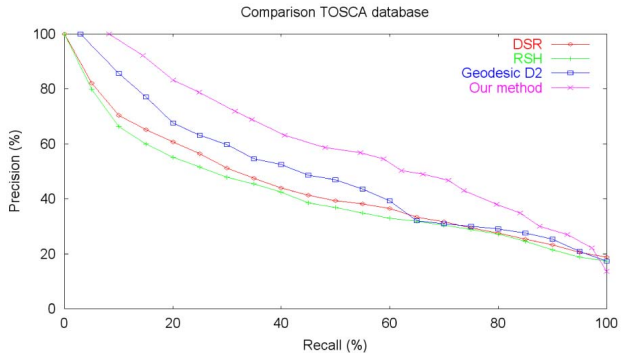


Fig. 8. Precision versus Recall plots comparing our approach to the RSH, D2, and DSR algorithms on the TOSCA data set.

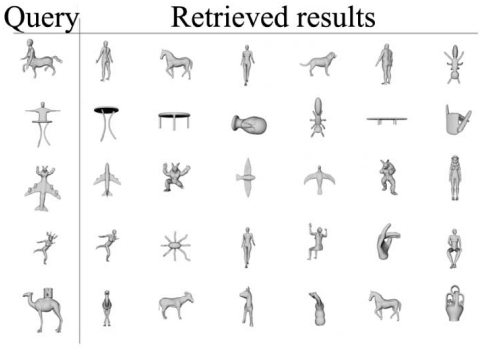


Fig. 9. First retrieved results on the SHREC07 data set.

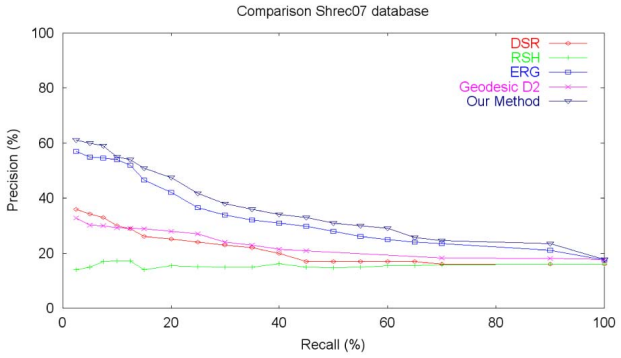


Fig. 10. Precision versus Recall plot on the SHREC07 data set.

5.5.2 Results on the SHREC07 Data Set

From a qualitative point of view, Fig. 9 gives a good overview of the efficiency of the framework. For example, in the first row, the query is a centaur, and thus, most of the top results are humanoid and animal models.

Quantitatively, we compare the Precision versus Recall plot of our approach with other methods competing in the contest. Such a plot is the average of the 30 Precision versus Recall plots corresponding to the 30 models of the query set. Fig. 10 shows these plots. As the curve of our approach is higher than the others, it is obvious that it outperforms related methods.

5.6 Robustness to Query Noises

We investigate the framework robustness against surface noise. For each element of the query set, we added a surface noise. The noise is $\pm 0.2\%$ and $\pm 0.5\%$ the length of the model's bounding box. It consists of three random translations in the three directions X, Y, and Z. Fig. 11 shows some examples of the noise effect.

Precision versus Recall plots are computed for our approach under the noise and compared with the original Precision versus Recall plot. From this, results shown in Fig. 12, the noise addition effects can be observed. These plots demonstrate the stability of the

Fig. 11. Surface noise on a TOSCA 3D model: original, $\pm 0.2\%$ noise and $\pm 0.5\%$ noise (from left to right).

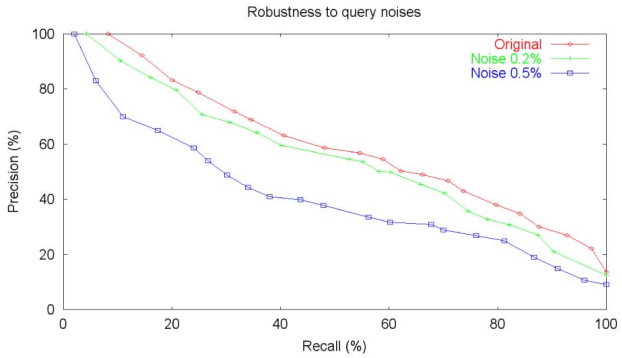


Fig. 12. Robustness to query noises on the TOSCA data set.

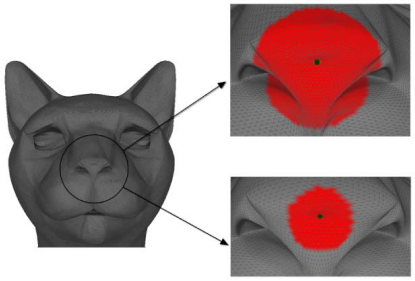


Fig. 13. An example of the feature point location with perturbation (1 percent bottom and 2 percent top).

algorithm despite random noise. Moreover, even with a random noise of $\pm 0.5\%$, it still outperforms the scores on clean data of the two methods presented in Fig. 8.

5.7 Robustness to Feature Points Localization

In order to assess the robustness of our method under localization of feature points, we apply the following transformation for the query set: We select a random vertex located in a geodesic radius (1 percent and 2 percent of the geodesic diameter of the 3D object) centered in the original feature points.

In Fig. 13, an example of the feature points localization with noise addition is shown. The original feature point is colored in green. The perturbation consists of a random selection of one vertex among vertices colored in red. Fig. 14 shows that the performance of our approach is robust with respect to 1 percent of perturbation. The 2 percent perturbation plot shows the limit of the algorithm in term of feature point perturbation. However, our method performance is still comparable to state-of-the-art method performance on clean data.

5.8 Discussion and Limitations

The feature point extraction step of the framework, which is based on geodesics, introduces a bias in the comparison process. To guarantee stability and performance, this method has to be stable within the same class of objects and, moreover, coherent with the data set ground truth. In practice, with the TOSCA and the SHREC07 data sets, feature extraction turns out to be homogeneous within most classes. However, the geodesic metric is known to be very sensitive to topology, which makes the overall framework also quite sensitive to topology variations. Fig. 15 shows the same query under topological change. In the first row, when the query genus is one, the system performs

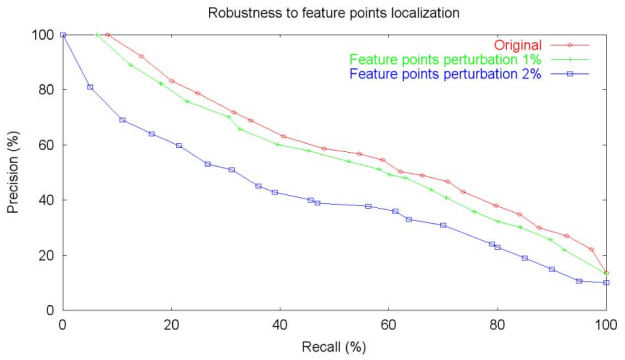


Fig. 14. Robustness to feature points localization on the TOSCA data set.

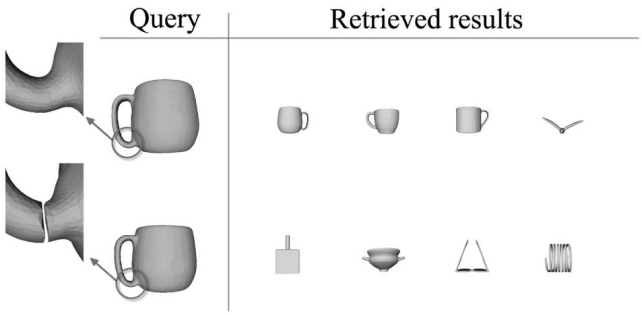


Fig. 15. First retrieved results of a 3D query with topological noise addition (the original query is on the top).

retrieval effectively. However, the second query, which represents the same cup with added topological noise (genus is zero), shows different results. In the future, we would like to investigate other decomposition strategies that would overcome this issue using more robust metrics such as diffusion or commute time distances [27] or using methods from 2D shape analysis for finding interest points and choosing features [28]. Moreover, because of the use of curve descriptors, the 3D objects in the database must be of quite high quality (manifold and high resolution).

6 Conclusion

In this paper, we have presented a geometric analysis of 3D shapes in the presence of nonrigid transformation and partially similar models. In this analysis, the preprocessing is completely automated: The algorithm processes the 3D object, extracts a set of feature points, and extracts a set of curves around each feature point to form a part. A 3D-part matching method is then presented to allow the construction of geodesic paths between two arbitrary 3D-part surfaces. The length of a geodesic path between any two 3D-part surfaces is computed as the geodesic length between the set of their curves. This length quantifies the differences in their shapes. Moreover, we have presented a 3D-shape matching method for global 3D search and retrieval applications. Belief functions has been used in this matching framework to compute a global similarity metric based on the 3D-part matching. The results on the TOSCA and the SHREC07 data sets show the effectiveness of our approach. They also show the robustness to rigid and nonrigid transformations, surface noise, and partially similar models. For future work, we plan to extend our approach for the detection of intrinsic symmetries in 3D objects.

Acknowledgments

The authors would like to thank Shantanu H. Joshi for providing a MATLAB code for computing a geodesic path between curves.

This work was partially supported by the project ANR-07-SESU-004 and the Contrat de Projet Etat-Région (CPER) Région Nord-Pas-de-Calais Ambient Intelligence.

REFERENCES

[1] G. Antini, S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval of 3D Objects Using Curvature Correlograms," *Proc. IEEE Int'l Conf. Multimedia and Expo,* 2005.

[2] T. Filali Ansary, M. Daoudi, and J.-P. Vandeborre, "A Bayesian 3D Search Engine Using Adaptive Views Clustering," *IEEE Trans. Multimedia,* vol. 9, no. 1, pp. 78-88, Jan. 2007.

[3] R. Gal, A. Shamir, and D. Cohen-Or, "Pose Oblivious Shape Signature," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 2, pp. 261-271, Mar./Apr. 2007.

[4] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes," *Proc. ACM SIGGRAPH,* pp. 203-212, 2001.

[5] D. Aouada, D.-W. Dreisigmeyer, and H. Krim, "Geometric Modeling of Rigid and Non-Rigid 3D Shapes Using the Global Geodesic Function," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition Workshops,* 2008.

[6] J. Tierny, J.-P. Vandeborre, and M. Daoudi, "Partial 3D Shape Retrieval by Reeb Pattern Unfolding," *Computer Graphics Forum,* vol. 28, pp. 41-55, 2009.

[7] Y. Liu, H. Zha, and H. Qin, "Shape Topics: A Compact Representation and New Algorithms for 3D Partial Shape Retrieval," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2006.

[8] A. Bronstein, M. Bronstein, A. Bruckstein, and R. Kimmel, "Partial Similarity of Objects, or How to Compare a Centaur to a Horse," *Int'l J. Computer Vision,* vol. 84, no. 2, pp. 163-183, 2008.

[9] S. Jahanbin, H. Choi, Y. Liu, and A.-C. Bovik, "Three Dimensional Face Recognition Using Iso-Geodesic and Iso-Depth Curves," *Proc. Second IEEE Int'l Conf. Biometrics: Theory, Applications and Systems,* 2008.

[10] C. Samir, A. Srivastava, M. Daoudi, and E. Klassen, "An Intrinsic Framework for Analysis of Facial Surfaces," *Int'l J. Computer Vision,* vol. 82, pp. 80-95, 2009.

[11] M. Mortara and G. Patanè, "Affine-Invariant Skeleton of 3D Shapes," *Proc. Int'l Conf. Shape Modeling and Applications,* pp. 245-252, 2002.

[12] S. Katz, G. Leifman, and A. Tal, "Mesh Segmentation Using Feature Point and Core Extraction," *The Visual Computer,* vol. 25, pp. 865-875, 2005.

[13] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, "Loops in Reeb Graphs of 2-Manifolds," *Proc. 19th Ann. Symp. Computational Geometry,* pp. 344-350, 2003.

[14] X. Li, Y. He, X. Gu, and H. Qin, "Curves-On-Surface: A General Shape Comparison Framework," *Proc. IEEE Int'l Conf. Shape Modeling and Applications,* 2006.

[15] Y. Laurent, "Computable Elastic Distance between Shapes," *SIAM J. Applied Math.,* vol. 58, pp. 565-586, 1998.

[16] P.W. Michor and D. Mumford, "Riemannian Geometries on Spaces of Plane Curves," *J. European Math. Soc.,* vol. 8, pp. 1-48, 2006.

[17] S. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "Removing Shape-Preserving Transformations in Square-Root Elastic (SRE) Framework for Shape Analysis of Curves," *Proc. Sixth Int'l Conf. Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 387-398, 2007.

[18] A.-J. Yezzi and A. Mennucci, "Conformal Metrics and True "Gradient Flows" for Curves," *Proc. 10th IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 913-919, 2005.

[19] E. Klassen, A. Srivastava, "Geodesics between 3D Closed Curves Using Path-Straightening," *Proc. Ninth European Conf. Computer Vision,* pp. 95-106, 2006.

[20] P. Smets and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence,* vol. 66, pp. 191-234, 1994.

[21] G. Shafer, *A Mathematical Theory of Evidence.* Princeton Univ. Press, 1976.

[22] A. Bronstein, M. Bronstein, and R. Kimmel, "Efficient Computation of Isometry-Invariant Distances between Surfaces," *SIAM J. Scientific Computing* , vol. 28, pp. 1812-1836, 2006.

[23] S. Marini, L. Paraboschi, and S. Biasotti, "Shape Retrieval Contest 2007: Partial Matching Track," *Proc. SHREC in Conjunction with Proc. Int'l Conf. Shape Modeling and Applications,* pp. 13-16, 2007.

[24] D. Saupe and D. Vranic, "3D Model Retrieval with Spherical Harmonics and Moments," *Pattern Recognition,* Springer, pp. 392-397, 2001.

[25] D. Vranic, "3D Model Retrieval," PhD dissertation, Univ. of Leipzig, May 2004.

[26] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape Distributions," *ACM Trans. Graphics,* vol. 21, no. 4, pp. 807-832, 2002.

[27] A. Bronstein, M. Bronstein, M. Mahmoudi, R. Kimmel, and G. Sapiro, "A Gromov-Hausdorff Framework with Diffusion Geometry for Topologically-Robust Non-rigid Shape Matching," *Int'l J. Computer Vision,* vol. 89, pp. 266-286, 2010.

[28] A. Gil, O.M. Mozos, M. Ballesta, and O. Reinoso, "A Comparative Evaluation of Interest Point Detectors and Local Descriptors for Visual Slam," *Machine Vision and Applications,* vol. 21, pp. 905-920, 2009.

Titre Contributions à la recherche et à l'analyse de modèles 3D

Résumé L'utilisation de modèles tridimensionnels dans les applications multimédia, prend de l'ampleur de jour en jour. Le développement des outils de modélisation, des scanners 3D, des cartes graphiques accélérées, du Web3D, etc. ouvre l'accès à des données tridimensionnelles de grande qualité. Les besoins, sans cesse croissants, concernant ce type de données, changent rapidement. S'il devient de plus en plus facile de créer de nouveaux modèles 3D, qu'en est-il du traitement et de l'analyse de ces modèles après leur création ? De nos jours, le concepteur d'objets 3D ne pose plus la question : « Comment créer un nouvel objet 3D ? », mais plus vraisemblablement « Comment retrouver un modèle 3D similaire à ceux en ma possession pour le réutiliser ? » et « Comment retrouver la structure d'un modèle 3D maillé sans connaissance a priori sur celui-ci ? »
Cette habilitation a pour but d'apporter des éléments de réponse à ces deux questions. En réponse à la première question, nous avons développé un nouveau système bayésien pour retrouver des modèles 3D à partir d'une requête constituée d'une ou plusieurs vues 2D, ou d'un modèle 3D entier. Ce système a été testé dans un contexte applicatif industriel ainsi qu'avec un benchmark international. Chaque expérience a mis en évidence les excellents résultats de notre approche. La seconde question a été abordée sous l'angle de l'analyse topologique des maillages 3D grâce aux graphes de Reeb. Ce travail théorique a été appliqué à différents domaines comme la déformation automatique, l'indexation et la segmentation de maillages 3D. L'approche a toujours montré des résultats remarquables dans ces domaines. Finalement, la segmentation de maillages 3D, qui est une étape de prétraitement fréquente avant d'autres analyses du maillage, a attiré notre attention. Nous avons proposé une métrique fiable et robuste pour la comparaison de segmentations et l'évaluation des performances des méthodes de segmentation de maillages 3D, ainsi qu'une approche de la segmentation par apprentissage qui surpasse les méthodes existantes. Pour terminer, de nouvelles pistes de recherche sur les maillages 3D sont ouvertes.

Mots-clés modèles 3D ; maillages ; indexation ; recherche ; analyse ; topologie ; segmentation

Title Contributions to 3D-model retrieval and analysis

Abstract The use of three-dimensional models in the multimedia applications, is growing both in number and in size. The development of modeling tools, 3D-scanners, graphic accelerated hardware, Web3D and so on, offers access to three-dimensional materials of high quality. The constantly increasing needs concerning these kinds of data, are rapidly changing. While it becomes more and more easy to create new 3D-models, what about process and analysis after the creation of the 3D-models ? Today, the 3D designer no longer asks : "How to create a new 3D-model ?", but more probably "How to retrieve an existing 3D-model that is similar to those I already own in order to reuse it ?" and "How to get the inner structure of a 3D-mesh model without any a priori knowledge on it ?"
This habilitation thesis aims to provide some answers to these two questions. In response to the first question, we developed a new Bayesian framework to retrieve 3D-models from a query made of one or more 2D-views, or of an entire 3D-model. The framework has been tested in an industrial application context and with an international benchmark. Each of these experiments has shown excellent results. The second question has been addressed in terms of topological analysis of the 3D-meshes with the help of Reeb graphs. This theoretical work has been applied to several practical domains, such as automatic 3D-mesh deformation, 3D-model retrieval, and 3D-mesh segmentation, and has always highlighted outstanding results. Finally, the segmentation of 3D-meshes, which is a frequent pre-processing step before any other analysis of the mesh, has drawn our attention. We proposed a reliable and robust metric to compare segmentations and evaluate the performances of the 3D-mesh segmentation methods, as well as a new learning-based segmentation approach that outperforms existing ones. To conclude, new perspectives of research on 3D-meshes are open.

Keywords 3D-models ; meshes ; indexing ; retrieval ; analysis ; topology ; segmentation