HAL open science

Recherches en reconstruction 3D photométrique

Emmanuel Prados

To cite this version:

Emmanuel Prados. Recherches en reconstruction 3D photométrique. Vision par ordinateur et reconnaissance de formes [cs.CV]. Université Joseph-Fourier - Grenoble I, 2012. tel-00747867

HAL Id: tel-00747867

https://theses.hal.science/tel-00747867

Submitted on 8 Nov 2017

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.

Synthèse des travaux et activités scientifiques

présentée par

Emmanuel Prados

pour obtenir le grade de
Habilitation à diriger des Recherches
de l'**UNIVERSITE DE GRENOBLE**
Spécialité : **Informatique et Mathématiques Appliquées**

Recherches en reconstruction 3D photométrique

Présentée publiquement le 04 AVRIL 2012

devant le jury composé de

Président :	James L. Crowley	INPG/INRIA Grenoble, PRIMA
Rapporteurs :	Jean-Denis Durou	Université Paul Sabatier Toulouse III, IRIT
	Marie-Odile Berger	INRIA Nancy, Magrit
	Michel Dhome	CNRS, LASMEA
Examinateurs :	Radu Horaud	INRIA Grenoble, Perception
	Olivier Faugeras	INRIA Sophia Antipolis, NeuroMathComp
	Patrick Criqui	CNRS, EDDEN

Remerciements

Tout d'abord je voudrais remercier tous les membres du jury qui ont gentiment accepté de participer à cet événement, en particulier ceux qui ont dû traverser la France pour arriver jusqu'à Grenoble. Merci à Jim d'avoir accepté de présider le jury ainsi qu'à Marie-Odile, Michel et Jean-Denis d'avoir accepté de jouer le rôle de rapporteurs. Un merci tout particulier aussi à Patrick qui a accepté de participer à ce jury malgré que l'essentiel des sujets abordés ne fasse pas parti de son domaine de prédilection, loin de là ! Merci à Marie-Anne pour son aide pour l'organisation de cet évènement, et pour son aide au quotidien.

Je voudrais aussi profiter de cette occasion pour remercier toutes les personnes qui m'ont soutenu ces quatre dernières années. Monter une nouvelle équipe de recherche dans un domaine d'application quasi vierge répondant à nos intérêts et convictions, ceci en ayant carte blanche, est une expérience grandiose mais, en même temps, particulièrement éprouvante (d'autant plus quand, initialement, on n'a aucune compétence dans le domaine). Et, sans le soutien d'un grand nombre de personnes, je ne serai jamais arrivé jusque là. Je voudrais remercier en particulier Alain Viari qui a soutenu mon projet dès le début, ainsi que François Sillion qui nous a offert un cadre idéal et les moyens de le mener à bien. Merci également à Hélène Kirchner pour son soutien. Merci à Radu qui a été particulièrement compréhensif et qui a tout fait pour nous faciliter les choses. Merci à Peter qui m'a soutenu dès le départ et qui nous a rejoint l'année dernière dans cette aventure. Un immense merci aussi a Patrick et Pierre-Yves qui m'ont ouvert leur porte et m'ont fait confiance dès nos premières rencontres et qui ont joué un grand rôle dans les choix scientifiques faits dans STEEP. Enfin, je voudrais particulièrement remercier Elise qui a été la première à véritablement s'engager et me suivre dans cette aventure et qui partage au quotidien et, en direct, déboires et satisfactions. J'en profite aussi pour remercier Olivier sans qui je ne serai jamais arrivé jusqu'ici et qui a, sans le savoir, influencé une partie de mes choix. A cet instant, je pense aussi à tous les étudiants que j'ai pu encadrer depuis 2006, en particulier à Pau et Amaël avec qui j'ai passé des instants vraiment passionnants.

Pour finir, merci à Laëtitia, Thomas, Lou et Eléa avec qui je partage aussi cette aventure, et surtout, avec qui je partage tout le reste. Sans eux ma vie n'aurait tout simplement pas le même sens…

Avant-propos

Dans ce document, je décris mes activités professionnelles durant la période allant de 2006 à 2011.

Après m'être exclusivement consacré pendant mes trois années de thèse (soutenue en 2004) au problème du "Shape From Shading" (reconstruction 3D monoculaire) et aux solutions de viscosité, j'ai effectué un post-doctorat d'un an à l'UCLA (2005) pendant lequel j'ai eu une activité scientifique assez diversifiée (segmentation en imagerie médicale, reconstruction de fibres en IRM-D, reconstruction 3D multi-vues,...). Ce document ne prend pas en compte ces quatre premières années de recherche et démarre donc en 2006, année de mon recrutement à l'INRIA R.-A. dans l'équipe PERCEPTION.

L'équipe PERCEPTION travaille sur la reconstruction de scènes 3D à partir d'images multi-vues, thématique autour de laquelle je me suis beaucoup amusé. Mais poussé par mes convictions je ressens très rapidement le besoin de travailler sur des thématiques plus en lien avec des préoccupations personnelles, en particulier, celles touchant au développement durable. Dès 2008, je me suis alors investi dans l'élaboration d'un nouveau projet de recherche qui a abouti en 2010 à la création de l'équipe INRIA STEEP. L'équipe STEEP travaille sur la modélisation des interactions entre économie, environnement et société dans le cadre de la mise en œuvre du développement durable à l'échelle locale (de la ville à la région). Les objectifs et activités de STEEP sont décrits dans la partie I de ce document (section I.2).

A cause de la jeunesse de l'équipe STEEP, les résultats scientifiques décrits dans ce document se focalisent sur mes contributions en vision par ordinateur (section I.2). La transition de mes activités de recherche de la vision par ordinateur aux modèles socio-économiques et environnementaux dans le cadre du développement durable a été longue et irrégulière. Les choses ont beaucoup d'inertie et inversement mettre en route quoi que ce soit de complètement nouveau nécessite toujours de dépenser beaucoup d'énergie.

La partie décrivant mes résultats scientifiques est précédée d'une description de mes activités d'animation de la recherche (organisations de colloques, participations à des comités de programme, responsabilités scientifiques, participations à des projets, communications en tant qu'invité, etc.) et des tâches administratives dont j'ai eu la charge (section I.1). La deuxième partie de ce document est une compilation de mes articles les plus représentatifs de mes activités de recherche pour la période concernée (partie II).

Table des matières

I Synthèse des activités scientifiques et administratives, et synthèse des travaux scientifiques **1**

1 Activités scientifiques et administratives **3**

1.1 Encadrement 3

1.1.1 Thèses 3

1.1.2 Post-docs 4

1.1.3 Diplômes de Recherche Technologique, stagiaires et visiteurs 5

1.2 Comités de programme, organisations de colloques, activités de relecture et autres responsabilités scientifiques 6

1.3 Responsabilités administratives 7

1.4 Participation à des projets 8

1.5 Communications en tant qu'invité 9

2 Synthèse des travaux scientifiques **11**

2.1 Synthèse des travaux scientifiques effectués en vision par ordinateur 12

2.1.1 Prise en compte et exploitation des contours en reconstruction 3D multi-vues 12

2.1.2 Fusionner les différentes informations d'ombrage, de contours et la correspondance 15

2.1.3 De la représentation "*level set*" aux maillages 16

2.1.4 Segmentation convexe sur des maillages 23

2.1.5 S'affranchir des modèles paramétriques tout en exploitant les propriétés de la réflectance 27

2.1.6 Retour aux sources : Le "*Shape From Shading*" 30

2.2 Brève description du projet scientifique de l'équipe STEEP 33

2.2.1 Développement de modèles intégrés systémiques prenant en compte les interactions entre environnement, économie et société 33

2.2.2 Calibration semi-automatique des modèles intégrés 34

2.2.3 Prise en compte des incertitudes et analyse de sensibilité 35

2.2.4 Apporter des éléments de réponse aux questions soulevées par le développement durable 36

Bibliographie complète d'Emmanuel Prados **43**

Références **46**

II Recueil d'articles **47**

Paper 1 : Minimizing the Reprojection Error in Surface Reconstruction from Images, ICCV 2007 51

Paper 2 : Joint Estimation of Shape and Reflectance using Multiple Images with Known Illumination Conditions, International Journal of Computer Vision, 2010 59

Paper 3 : Gradient Flows for Optimizing Triangular Mesh-based Surfaces : Applications to 3D Reconstruction Problems dealing with Visibility, International Journal of Computer Vision, 2011 79

Paper 4 : Convex Multi-Region Segmentation on Manifolds, ICCV 2009 111

Paper 5 : Towards Full 3D Helmholtz Stereovision Algorithms, ACCV 2010 119

Paper 6 : A Non-Local Approach to Shape From Ambient Shading, SSVM'09 133

Paper 7 : Modélisation numérique : Quel développement durable ?, La Recherche, oct 2010 145

III Conclusion **151**

Première partie

Synthèse des activités scientifiques et administratives, et synthèse des travaux scientifiques

Chapitre 1

Activités scientifiques et administratives

1.1 Encadrement

J'ai encadré et co-encadré une vingtaine de personnes jusqu'à ce jour, dont trois doctorants. Deux thèses ont déjà été soutenues, en 2006 et 2011. Une est toujours en cours (fin de première année). Les encadrements sont décrits plus en détail dans la suite. La plupart de mes travaux scientifiques depuis cinq ans ont été menés *via* ces encadrements.

1.1.1 Thèses

Pau Gargallo Doctorant **PERCEPTION,** (2003-07), co-encadrement (15 %) avec Peter Sturm (85 %). Directeur de thèse : Peter Sturm. Plus exactement, j'ai co-encadré Pau pendant sa dernière année de thèse (2006-2007). Titre de la thèse : *Contributions to the Bayesian Approach to Multi-View Stereo*. Pau et moi avons travaillé en particulier sur la prise en compte des contours et de la visibilité dans les problèmes de reconstruction de surfaces 3D à partir d'images multi-vues. Notre analyse montre la forte influence du mouvement des générateurs de contours dans le gradient de l'erreur de reprojection ce qui permet de déplacer automatiquement ces générateurs de contours vers leur emplacement correct dans les images. Ce travail permet aussi de mieux comprendre et de justifier les méthodes de l'état de l'art assurant cet alignement via des contraintes additionnelles de silhouettes ou de contours apparents. Ici, la difficulté était d'arriver à prendre en compte correctement les changements de visibilité qui se produisent lorsque la surface se déplace. Les travaux effectués avec Pau ont notamment été publiés à ICCV 2007 [10] ; voir aussi [9, 8, 48, 11, 54].

Amaël Delaunoy Doctorant **PERCEPTION,** (2007-11), co-encadrement (80 %) avec Peter Sturm (20 %). Directeur de thèse : Peter Sturm. Titre de la thèse : *Multi-view Shape Reconstruction from Images : Contributions Towards Generic and Practical Solutions using Deformable Meshes*. Le travail d'Amaël repose sur les mêmes paradigmes que ceux de Pau Gargallo et de Kuk-Jin Yoon (c.f. ci-dessous), c'est à dire que les problèmes de reconstruction 3D pourraient être nettement mieux posés et résolus si nous arrivions à intégrer et exploiter simultanément un maximum d'informations et de contraintes ("*cues*"), en particulier celles liées aux ombrages, à la correspondance et aux contours. Ayant décelé un certain nombre de

difficultés avec les méthodes "*level-set*" sur lesquelles se basent les travaux que j'ai effectués avec Kuk-Jin et Pau, Amaël a tout d'abord étendu ces travaux aux maillages. Par ailleurs, de façon à pouvoir exploiter la redondance de l'information présente dans une scène faite d'un nombre fini de matériaux, Amaël a montré comment utiliser des méthodes de relaxation convexe pour segmenter la surface (maillage triangulé) en régions ayant les mêmes propriétés (albédo, radiance ou réflectance). Enfin, il montre comment faire des reconstructions 3D complètes d'objets ayant des propriétés de réflectance arbitraires en exploitant la réciprocité de Helmholtz. Les travaux effectués avec Amaël ont abouti à de nombreuses publications [5, 9, 6, 15, 7, 4, 8, 48].

Le travail de thèse d'Amaël a été récompensé à deux reprises. Il lui a été attribué le **prix de la meilleure thèse AFRIF** (Association Française pour la Reconnaissance et l'Interprétation des Formes). Nous avons aussi obtenu un des prix attribués lors de la conférence BMVC 2008 : le **prix "CRS Industrial Prize"**.

1.1.2 Post-docs

Kuk-Jin Yoon, Post-doctorant **PERCEPTION,** (2006-08), co-encadrement (80 %) avec Peter Sturm (20 %). Avec Kuk-Jin, nous avons proposé un modèle de reconstruction 3D qui permet de récupérer à la fois la forme et la réflectance de la surface d'une scène à partir d'images multiples, en supposant que les conditions d'éclairage et la calibration des caméras sont connues à l'avance. La méthode proposée est très générale et s'applique indistinctement à un certain nombre de scénarios classiques, en particulier, il s'applique à la stéréovision, à la stéréovision photométrique multi-vues ainsi qu'au "*shape from shading*" multi-vues. Notre approche combine naturellement dans un cadre unique les informations de correspondance, de silhouette et d'ombrage. Par ailleurs, contrairement à la plupart des méthodes précédentes pouvant traiter uniquement des surfaces lambertiennes, la méthode proposée s'applique de manière générale à toute surface dichromatique bien approximée par un modèle de réflectance paramétrique connu à l'avance. Les travaux effectués avec Kuk-Jin ont abouti aux publications suivantes : [53, 48, 52, 54].

Zsolt Janko, Post-doctorant **PERCEPTION,** (2009), encadrement à 100 %. Zsolt a travaillé sur la stéréovision photométrique, un cadre expérimental spécifique qui permet d'exploiter les propriétés de réflectance d'une scène illuminée par plusieurs configurations d'éclairage, afin d'en reconstruire la forme 3D. Jusqu'à très récemment, la stéréovision photométrique était restreinte aux scènes statiques. Un des objectifs du travail de Zsolt était d'étendre ce cadre aux scènes dynamiques. Les travaux effectués avec Zsolt Janko ont été publiés à ACCV'10 [15].

Subhash Mallah, Post-doctorant **STEEP,** (2011-13), co-encadrement (40 %) avec Pierre-Yves Longaretti (20 %) et Patrick Criqui (40 %). Subhash est arrivé chez STEEP courant 2011. L'objectif de son travail est de développer un modèle intégré transport-usage des sols-énergie pour la ville de Grenoble en se basant sur les modèles TRANUS (transport et usage des sols) et ETEM (énergie) ainsi que sur les données et analyses produites dans le cadre du projet ANR AETIC coordonné par Patrick Criqui.

Parikshit Dutta, Post-doctorant **STEEP,** (2011-12), co-encadrement (20 %) avec Elise Arnaud (40 %) et Peter Sturm (40 %). Parikshit est arrivé dans l'équipe STEEP pendant l'été 2011. Il travaille sur la gestion des incertitudes et sur l'analyse de sensibilité pour les modèles de transport-usage des sols, en particulier le modèle TRANUS.

1.1.3 Diplômes de Recherche Technologique, stagiaires et visiteurs

Claude Mergen, Diplôme de Recherche Technologique (**STEEP,** 2009-10), co-encadrement (30 %) avec Elise Arnaud (30 %) et Alban Kitous, Enerdata (40 %). Directeur de thèse : Eric Blayo. Le DRT est une sorte de mini thèse professionnelle d'un an. Pendant cette année, Claude a appliqué le modèle d'énergie ETEM à la ville de Grenoble.

Stagiaires Pendant mes quatre années dans l'équipe Perception, j'ai encadré quatre stagiaires

PERCEPTION 2006 → 2008 : Gaurav Bubna, (2007, Master 1), Ravi Garg, Internship (2008, Master 2), Julie Escoda, MSc project (2008, Master 2), Nitin Jindal (2006, Master 1 et 2007, Master 2).

En particulier, les deux stages de Nitin Jindal ont abouti à la publication d'un article de conférence publié à SSVM'09 [43].

Dans le cadre des activités de l'équipe STEEP, j'ai encadré et co-encadré une dizaine de stages avec Elise Arnaud et Pierre-Yves Longaretti :

STEEP 2010 : Abhishek Upadhyay (Master 1), Chao Wang (Master 1), Marie Chevalier (L3), Emmanuel Iarussi (Master 2), Enzo Ferrante (Master 2),

STEEP 2011 : Mariano Fernandez (Master 2), Alejandro Deymonnaz (Master 2), Hugo Luis Manterola (Master 2), Mariano Fernandez (Master 2), Anthony Tschirhard (Master 1).

Visiteurs

Ketut Fundana, doctorant de l'université de Malmö, Suède. Visite de six mois dans le cadre de **PERCEPTION,** entre 2008 et 2009. Ketut a travaillé avec Amaël et moi sur la segmentation multi-régions de données sur les surfaces, voir section 2.1. Cette collaboration a abouti à la publication d'un article à ICCV'09 [4] ; voir aussi [7].

1.2 Comités de programme, organisations de colloques, activités de relecture et autres responsabilités scientifiques

Activités en vision par ordinateur :
Ces activités s'étalent principalement sur la période 2006 - 2009. Depuis 2009, par manque de temps et vue ma ***nouvelle orientation scientifique***, j'ai dû refuser presque systématiquement toutes les invitations à être membre de comités pour des évènements ou conférences de vision par ordinateur.

Comités de programme et organisations de colloques :

ECCV 2008 J'ai été *Tutorials Chair* de la conférence ECCV 2008 (*European Conference on Computer Vision*) organisée à Marseille, France, en octobre 2008.
http://eccv2008.inrialpes.fr/

3DFP'08 J'ai été membre du comité de programme de 3DFP'08 (*3D Face Processing Workshop*), organisé lors de la conférence CVPR 2008 à Anchorage en Alaska le 27 juin 2008.
http://www.cs.york.ac.uk/3dfp/

SciCADE 2007 J'ai organisé un symposium intitulé "*PDEs and image processing*" au SciCADE 2007 (*International Conference on SCIentific Computation And Differential Equations*) qui a eu lieu en juillet 2007 à Saint-Malo, France.
http://scicade07.irisa.fr/

Congrès SMAI 2007 J'ai été l'organisateur du symposium intitulé "*variational and PDE methods for computer vision and image processing*" en conjonction avec le Congrès SMAI 2007 organisé à Praz sur Arly, France, en juin 2007.
http://www-ljk.imag.fr/smai2007/

PACV'07 J'ai été organisateur et *General Chair* du *Workshop* PACV 2007 (*Photometric Analysis for Computer Vision*) organisé lors de la conférence ICCV 2007 (*International Conference on Computer Vision*) qui a eu lieu à Rio de Janeiro au Brésil en octobre 2007.
http://iccv2007.rutgers.edu/
http://pacv2007.inrialpes.fr/

SSVM'07 J'ai été membre du comité de programme de la conférence SSVM'07 (*Scale Space and Variational Methods Conference*), organisé à Ischia en Italie en juin 2007.
http://ssvm07.ciram.unibo.it/ssvm07_public/

VLSM'05 J'ai été membre du comité de programme de VLSM'05 (*Workshop on Variational, Geometric and Level Set Methods in Computer Vision*), organisé lors de la conférence ICCV 2005 à Beijing en Chine en octobre 2005.

Relectures d'articles et éditeur invité : J'ai effectué des relectures d'articles pour les meilleures conférences et revues internationales en vision par ordinateur (en particulier International Journal Of Computer Vision, Transactions on Pattern Analysis and Machine Intelligence IEEE Transactions on Image Processing, Journal of Mathematical Imaging and Vision, Computer Vision and Image Understanding, Image and Vision Computing Journal, Pattern

Recognition Letters, etc et CVPR, ICCV, ECCV, VLSM, SSVM, etc).

Edition : J'ai été éditeur invité pour un numéro spécial du journal IJCV avec P. Belhumeur, K. Ikeuchi, S. Soatto et P. Sturm [1]. J'ai été éditeur des "*proceedings*" de la conférence PACV'07 organisé lors de ICCV 2007 à Rio de Janeiro [14]

Membre de jury de thèses :

- Mickael Péchaud, Université de Paris Diderot - Ecole Normale Supérieure - Ecole Nationale des Ponts et Chaussées, octobre 2009.
- Pau Gargallo, Université de Grenoble, janvier 2007.
- Amaël Delaunoy, Université de Grenoble, décembre 2011.

Activités en lien avec l'équipe STEEP :

Expert pour l'évaluation de projets :

- Membre du **comité de pilotage du programme phare de la FRB** ("Fondation pour la Recherche sur la Biodiversité") "Modélisation et scénarios pour la biodiversité", depuis juin 2010.
- Rapporteur pour l'ANR pour le programme "Chaires d'excellence" et pour le pôle Advan*city* : Green Technologies and sustainable cities cluster, en 2010.

Organisations de colloques :

Journée développement durable à l'INRIA J'ai été l'organisateur de la journée "développement durable à l'INRIA" intitulé "Le développement durable : les impacts sociétaux et défis scientifiques" qui a eu lieu le 22 Mars 2010 simultanément sur les huit sites INRIA en France (Grenoble, Saclay, Rocquencourt, Rennes, Nancy, Sophia Antipolis, Bordeaux et Lille).

Conférence SOCLE3 à l'IEP, Grenoble J'ai organisé la journée la conférence "SOCLE3" qui a eu lieu en février 2010 à l'Institut d'Etudes Politiques de Grenoble. http://socle3.obs.ujf-grenoble.fr/?p=218

1.3 Responsabilités administratives

- **Responsable de l'équipe STEEP** (équipe de l'INRIA Rhône-Alpes et du Laboratoire Jean Kuntzmann) depuis sa création en janvier 2010.
- Responsable informatique (CMI) de l'équipe PERCEPTION de 2006 à 2010 (parc d'une quarantaine d'ordinateurs).

1.4 Participation à des projets

FLAMENCO (Modélisation de scène dynamiques), projet ANR – Agence Nationale de la Recherche, du programme MDCA 2006.
J'ai été **coordinateur** du projet Flamenco.
- Montant financé par l'ANR : 340 000 Euros.
- Durée : 4 ans ; 2007-2010.
- Partenaires : CERTIS (Ecole Nationale des Ponts et Chaussées) et l'équipe Perception (INRIA Rhône-Alpes).

Le but de ce projet est de s'attaquer aux verrous de la reconstruction de scènes dynamiques en prenant mieux en compte, entre autre, les aspects photométriques.

SOCLE3 (Vers des espaces urbains et périurbains durables), Projet du programme PIRVE 2009.
Je suis co-auteur et partenaire du projet SOCLE3.
- Montant financé : 20 000 Euros.
- Durée : 2 ans.
- Partenaires : LEPII, LGGE, OSUG, PACTE, INRIA (STEEP).
- Coordinateurs Pierre-Yves Longaretti (OSUG) et François Mancebo (PACTE).

P.-Y. Longaretti, Patrick Criqui, F. Mancebo et moi même sommes les fondateurs du groupe de recherche interdisciplinaire SOCLE3.

TRACER (TRanus, Analyse de la Calibration et des Erreurs, Retours sur Grenoble et Caracas), soumis au programme ECOS-NORD 2011.
Je suis co-auteur et partenaire du projet TRACER.
- Montant financé : 100 000 Euros.
- Durée : 4 ans.
- Partenaires : IDDRI (Institut du développement durable et des relations internationales), Modelistica (Venezuela, auteur du modèle TRANUS), STEEP (INRIA Rhône-Alpes).
- Coordinateur : IDDRI.

Le but de ce projet est de renforcer la collaboration entre les différents partenaires autour des questions de calibration du modèle de transport et usage de sols TRANUS.

ESNET (Futur des réseaux de services écologiques dans la région urbaine de Grenoble), soumis au programme Phare de la FRB (Fondation pour la Recherche en Biodiversité) "Modélisation et scénarios pour la biodiversité", 2011.
- Montant demandé : 270 000 Euros.
- Durée : 3 ans.
- Partenaires : LECA, LEPII, Cemagref, PACTE, ERIC, INRIA Rhône-Alpes (STEEP).
- Coordinateur : LECA.

MUTERA "Modèles Urbanisme-Transport-Environment en Rhône-Alpes", 2011–.
- Proposition d'un groupe de travail régional réunissant les agences et acteurs locaux de l'urbanisme et du transport de la région Rhône-Alpes ainsi que des chercheurs autour des modèles d'urbanisme, de transport et d'environnement.
- Partenaires : AURG, SMTC, LEPII, LET, IAU-IDF, IDDRI, ERIC, STEEP (INRIA Rhône-Alpes).
- Coordinateur : AURG.

1.5 Communications en tant qu'invité

J'ai effectué les communications suivantes, pour des séminaires invités ou en tant que conférencier invité :

- *MCDA72*, 72nd meeting of the European Working Group "Multiple Criteria Decision Aiding", Paris, France, 7-9 octobre 2010. Organisateur : Vincent Mousseau, Ecole Centrale Paris.
- *Séminaire du département "MAD"* du laboratoire Jean Kuntzmann : "Systemic modelling of the interactions between environment, economy and society at local scales : problems and bottlenecks. ", Grenoble, France, 19 avril 2011. Organisateur : Nicolas Papadakis.
- *Conférence franco-canadienne BABEL : "Are sustainable urban policies going to be reduced to global warming issues"*, Grenoble, France, 18-19 novembre 2009. Organisateur : François Mancebo.
- *Workshop "mathematical methods for image analysis"*, Orléans, France, avril 2008. Organisateur : Maïtine Bergounioux.
- *SciCADE 2007*, Saint-Malo, France, 9-13 juillet 2007, Organisateur : Philippe Chartier.
- *"GdR ISIS : 3D modelling from images"*, ENST, Paris, France, 15 novembre 2006. Organisateur : Peter Sturm.
- *Séminaire du LMC*. Grenoble, France, 16 novembre 2006. Organisateur : LMC (Laboratoire de Modélisation et Calcul de Grenoble).
- *Séminaire "Front propagation and applications" du CERMICS*. Marne la Vallée, France, 7 mars 2006. Organisateur : Régis MONNEAU (CERMICS).
- *"Congrès SPECIF 2006"*. Saint-Étienne, France, 12-13 janvier 2006.
- *"Optimization Methods in Computer Vision"*. Les Houches, France, mars 2006. Organisateurs : VISIONTRAIN.
- *Mathematics and Image Analysis 2004 (MIA'04)*. Paris, France, 6-9 septembre 2004. Organisateur : Laurent Cohen (Ceremade).
- Séminaire "Probability, Optimisation and Control". Paris, France, 28 octobre 2004. Organisateur : Maxplus Lab. (INRIA).
- Séminaire GRAVIR, Grenoble, France, 3 mars 2005.
- *Séminaire IPAM*. UCLA, USA, juin 2005. Organisateur : Luminita Vese.
- Invitations/Séminaires 2004/2005 : Projet Magritte, LORIA (Marie Odile Berger), Projet Anubis, INRIA Bordeaux (Jacques Henry), Projet MIRAGES, INRIA Rocquencourt (André Gagalowicz).

Chapitre 2

Synthèse des travaux scientifiques

Dans ce chapitre, je décris très succinctement mes activités de recherche depuis 2006. Les travaux décrits ici ont été effectués depuis mon recrutement à l'INRIA, en tant que Chargé de Recherche (mi-décembre 2005). Pour chaque thématique abordée, je citerai les noms des collaborateurs y ayant contribué.

Pendant mes premières années travaillées à l'INRIA (disons 2006-2010), j'ai travaillé dans le domaine de la vision par ordinateur, plus exactement en reconstruction 3D multi-vues. Très tôt, dès fin 2008, je me suis lancé dans l'élaboration d'un nouveau projet scientifique autour de la modélisation systémique des interactions entre environnement, économie et société afin de pouvoir développer des outils d'aide à la décision permettant de faciliter la mise en oeuvre de politique de développement durable ; ce qui a abouti à la création de l'équipe STEEP en 2010.

Dans ce chapitre, je décris dans un premier temps mes activités et résultats de recherche réalisés en vision par ordinateur. Je décris ensuite très brièvement le projet scientifique de l'équipe STEEP.

Comme il est dit dans l'avant-propos, cette synthèse est accompagnée à la fin du document d'un recueil de mes articles les plus représentatifs en rapport avec les travaux décrits. Les liens entre ces articles et les travaux synthétisés dans cette section sont faits au fil de l'eau. Dans cette synthèse je ne ferai pas d'état de l'art, car les articles rassemblés à la fin de ce document contiennent tous un état de l'art relativement détaillé des différents domaines étudiés. De la même façon, ce chapitre ne contiendra aucun détail technique et peu de résultats expérimentaux. Pour plus d'information, de détails et de résultats je réfère le lecteur aux articles joints dans la partie II.

2.1 Synthèse des travaux scientifiques effectués en vision par ordinateur

Pendant ces quatre années dans l'équipe Perception, mes travaux de recherche en vision 3D ont essentiellement été guidés par une conviction : pour arriver à progresser dans ce domaine, il est nécessaire d'arriver à aller vers plus d'intégration (des différentes informations disponibles dans les images) et vers plus de modélisation.

Les images contiennent un grand nombre d'informations ("*cues*"), en particulier la correspondance, l'ombrage et les contours. En reconstruction 3D multi-vues, toutes ces informations n'ont cependant été que très partiellement fusionnées. Pourtant en exploitant simultanément le maximum d'informations disponibles, nous devrions intuitivement obtenir de meilleurs résultats (en terme de précision) et des algorithmes plus robustes (capacité à gérer des situations critiques dans lesquelles certains types d'informations sont en carence). Par ailleurs, nous avons aussi régulièrement des connaissances *a priori* sur la scène (éclairage, scène constituée d'un nombre fini d'objets ou de matériaux, . . .) ; connaissances que l'on peut alors essayer d'utiliser, par exemple en les insérant sous la forme de contraintes. Un des fils conducteurs de mes recherches a donc toujours été d'arriver à trouver un cadre rigoureux permettant de mêler et exploiter naturellement et simultanément toutes ces informations pour les problèmes de reconstruction 3D multi-vues. Pour arriver à ça, je pense aussi qu'il est nécessaire de travailler et de se replonger dans la modélisation. Ces quinze dernières années, à raison, la communauté s'est essentiellement focalisée sur les aspects algorithmiques du problème ; ce qui a permis d'aboutir à des outils relativement sophistiqués, matures et performants. Les modèles utilisés pour représenter le processus de formation des images sont quant à eux restés très élémentaires. Pour aller plus loin, et pour pouvoir aussi intégrer d'avantage d'informations, il me semble alors inévitable de revenir sur ces questions.

2.1.1 Prise en compte et exploitation des contours en reconstruction 3D multi-vues

Avec Pau Gargallo, nous nous sommes alors penchés sur la question de la gestion des contours apparents et sur la prise en compte de la visibilité, notion qui lui est directement liée dans les problèmes de reconstruction 3D. Aussi, comme il est très largement fait dans la littérature (de manière implicite ou explicite) nous nous sommes naturellement dirigés vers des formulations sous la forme d'un problème d'optimisation.

En s'inspirant en particulier des travaux sur la segmentation stéréoscopique de Yezzi et Soatto [74], nous nous sommes rendus compte que la maximisation de la photo-cohérence entre les images d'entrée ("*input*") et les images générées à partir de la géométrie de la surface et de l'apparence estimées (critère correspondant au terme de vraisemblance dans la formulation bayésienne du problème) contenait en fait déjà l'information de visibilité. Ce critère, que nous avons appelé "erreur de reprojection", bien que le plus simple et le plus naturel qui soit, se trouve être extrêmement difficile à optimiser rigoureusement. Pour se faire, nous avons effectué une descente de gradient. Le problème devient alors comment calculer rigoureusement ce gradient en prenant en compte correctement les changements de visibilité lorsque la surface se déplace. En apportant une réponse à la question "Est ce que l'on voit un point du fond ou un point de la surface", les tra-

vaux de Yezzi et Soatto [74] permettent en somme de calculer un tel gradient pour des surfaces convexes. Dans [10], nous avons rigoureusement développé et analysé le gradient dans le cas général. Ce calcul rigoureux aboutit à l'apparition d'un nouveau terme dans le gradient qui correspond au mouvement des générateurs de contours (qui avait été négligé jusqu'à présent). Notre analyse montre alors la forte influence de ce mouvement dans le gradient de l'erreur de reprojection. Ce nouveau terme mis à jour permet en fait de déplacer automatiquement les générateurs de contours vers leur emplacement correct dans les images. Ce travail a permis ainsi au passage de mieux comprendre et de justifier (via l'erreur de reprojection) les méthodes de l'état de l'art assurant cet alignement via des contraintes additionnelles de silhouettes ou de contours apparents.

L'intérêt de ce travail, en particulier du nouveau terme de contours que nous avons ainsi obtenu dans le gradient est illustré dans les Figures 2.1 et 2.2.

Le premier jeu de données appelé "balles" (Figure 2.1) est constitué de 20 images de trois boules flottant sur un plan. Il n'y a pas de texture ni d'ombrage dans aucune partie de la scène. La seule information présente dans l'image sont donc les contours apparents. Par ailleurs, à cause des auto-occultations entre les boules et le plan, les silhouettes ne sont pas suffisantes pour distinguer que les boules sont trois objets séparés. Lorsque nous utilisons le gradient de l'erreur de reprojection dans son intégralité, l'algorithme réussit avec succès à obtenir une reconstruction correcte et à séparer les trois balles. Nous obtenons par ailleurs le même résultat en ne prenant en compte que le nouveau terme de contours ; c.f. Figure 2.1. Par contre, si dans le gradient nous utilisons uniquement le terme intérieur (ce qui est classiquement fait), l'algorithme ne sépare pas les boules durant le processus d'évolution et, à cause du manque de texture, la surface se rétracte et disparaît. Ce comportement reste le même lorsque l'on initialise la surface avec la surface à l'origine des données (le "*ground truth*").

La scène du "bol" illustrée dans la Figure 2.2 contient une balle verte à l'intérieur d'un bol jaune, le tout étant Lambertien. L'exécution avec le flot dans son intégralité reconstruit correctement la concavité du bol ainsi que la forme des boules. L'exécution utilisant seulement le terme d'horizon (nouveau terme) ne réussit pas du tout à creuser la concavité. L'exécution avec le terme intérieur (terme classique) réussit à creuser la concavité mais pas complètement, laissant des connexions entre la balle et le bol. Ceci montre comment le terme intérieur et le terme d'horizon agissent ensemble ; le premier creuse la concavité quand le second force l'alignement des générateurs de contours de la boule avec ses contours apparents sur les images.

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 1 [10]

▷ PUBLICATIONS ASSOCIÉES : [10, 11]

▷ COLLABORATEURS : Pau Gargallo (INRIA, équipe Perception) et Peter Sturm (INRIA, équipe Perception)

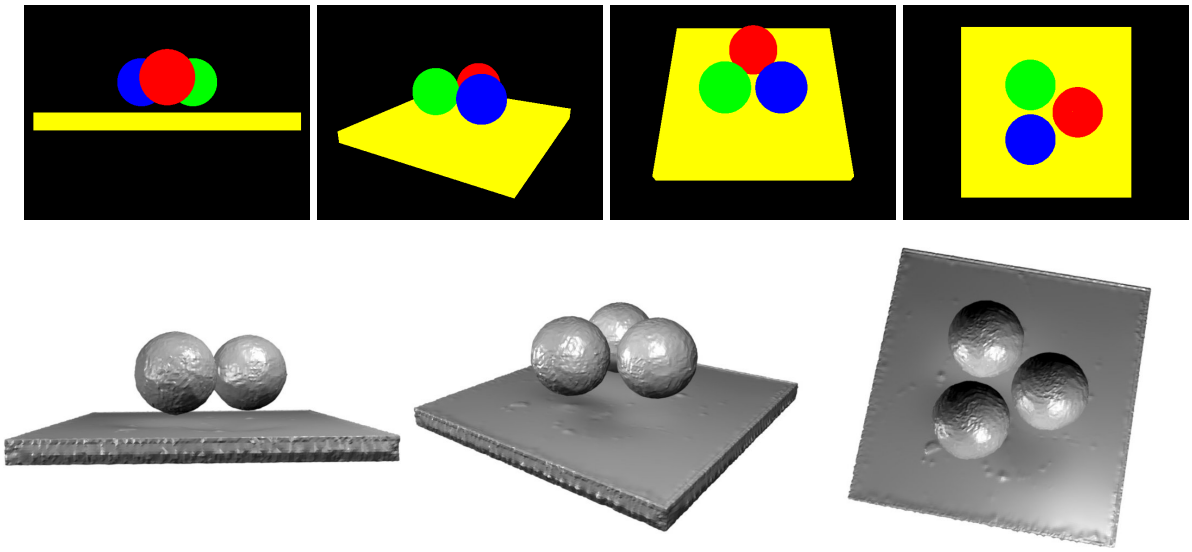


FIGURE 2.1 – Deux images d'entrée du jeu de données des "balles" et trois visualisations de la reconstruction obtenue avec le terme d'horizon.

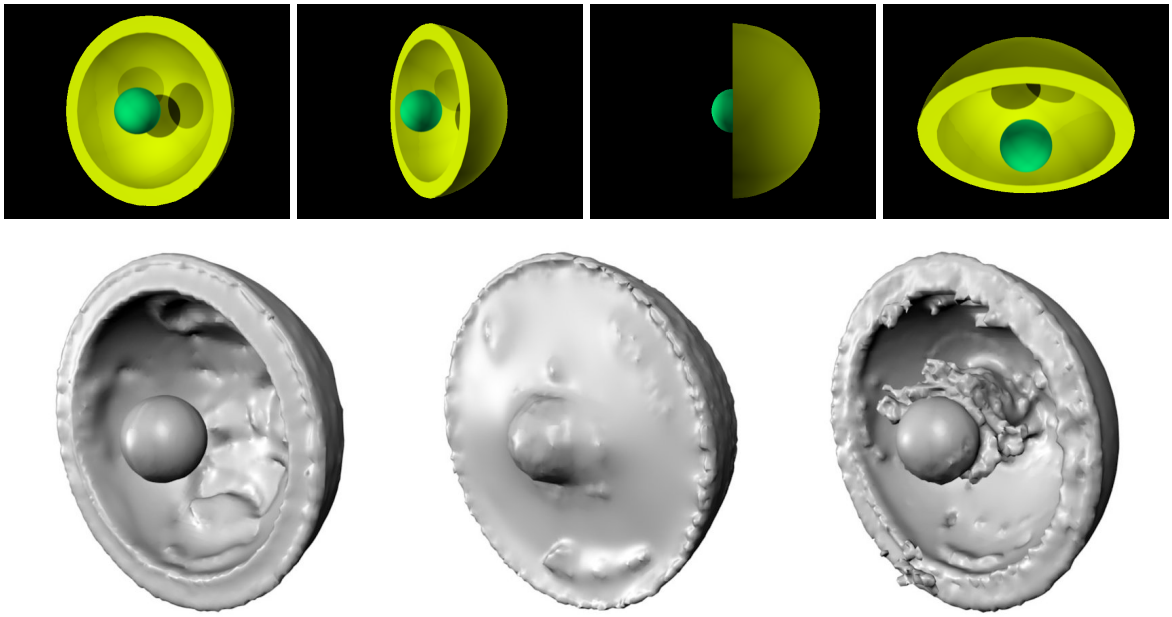


FIGURE 2.2 – Deux images d'entrée du jeu de données du "bol" et la reconstruction obtenue avec le gradient de l'erreur de reprojection en entier, avec juste le terme d'horizon (nouveau terme que nous avons mis au jour dans nos travaux) et enfin, avec juste le terme intérieur (terme classique de correspondance).

2.1.2 Fusionner les différentes informations d'ombrage, de contours et la correspondance

En y regardant de plus près, il paraît dès lors clair qu'en comparant directement les images d'entrée et les images générées par le modèle estimé, la minimisation de l'erreur de reprojection permet aussi de prendre en compte naturellement les effets de couleurs et d'ombrages ; ceci, à partir du moment où le modèle prend en compte les propriétés de réflectance et où nous sommes capables de générer des images avec ces mêmes effets.

Par conséquent, la minimisation de l'erreur de reprojection permet de fusionner naturellement stéréo, ombrage et contours dans un même cadre sans nécessiter l'introduction de paramètres de réglage pour moduler leurs influences relatives.

Avec Kuk-Jin Yoon, nous avons alors utilisé le cadre que nous avons déployé avec Pau, pour développer un algorithme de reconstruction 3D particulièrement générique permettant de reconstruire à la fois la forme et les propriétés de réflectance de la scène.

La méthode proposée s'applique indistinctement à un certain nombre de scénarios classiques. En particulier, il s'applique à la stéréovision, à la stéréovision photométrique (multi-vues ou non) ainsi qu'au "*shape from shading*" multi-vues. Comme espéré, notre algorithme démontre sa capacité à combiner naturellement un maximum d'informations (correspondance, contours et ombrage). Par ailleurs, contrairement à la plupart des méthodes précédentes qui s'appliquent uniquement aux surfaces lambertiennes, la méthode que nous proposons permet de gérer toute sorte de modèles paramétriques de réflectance.

Dans cet algorithme, nous avons par contre supposé que les conditions d'éclairage (modélisées par un terme ambiant et un nombre fini de sources de lumière ponctuelles ou directionnelles qui peuvent être différentes pour chacune des images) et la calibration des caméras sont connues à l'avance. La méthode suppose aussi qu'un modèle paramétrique de réflectance connu à l'avance (aux paramètres près qui seront alors estimés automatiquement par l'algorithme) représente relativement bien les propriétés de réflectance de la scène photographiée. Pour simplifier, dans l'algorithme implémenté nous avons supposé que ce modèle était un modèle de Blinn-Phong, mais celui-ci peut être changé sans aucune difficulté. Les divers coefficients diffus comme spéculaires peuvent varier et sont estimés en chacun des points de la surface. Pour arriver à bien poser le problème, de fortes hypothèses de régularité sont par contre nécessaires.

Nous avons appliqué notre approche aux différents scénarios mentionnés précédemment (stéréovision, stéréovision photométrique multi-vues, "*shape from shading*" multi-vues), avec des scènes texturées et non texturées, avec des surfaces lambertiennes et des surfaces montrant de forte réflexions spéculaires. Les Figures 2.3, 2.4 et 2.5 illustrent un certain nombre de résultats obtenus.

La Figure 2.3 montre un exemple de résultats obtenus avec une surface fortement mais uniformément spéculaire (coefficients diffus et spéculaires constants) prise sous des éclairages et des points de vue différents. La plupart des méthodes 3D multi-vues de l'état de l'art ne fonctionnent pas pour un ensemble d'images obtenues avec un éclairage variable. Par ailleurs, elles peinent en général à obtenir des résultats corrects en présence de fortes spécularités. Malgré la difficulté de l'exercice, l'algorithme que nous avons proposé arrive à reconstruire une surface 3D plus que raisonnable

(a) "*ground-truth*" (b) forme estimée (c) image correspondant à la composante diffuse estimée (d) image correspondant à la composante spéculaire estimée (e) image resynthétisée

FIGURE 2.3 – Résultats obtenus à partir d'un ensemble d'images de la "Bimba" (36 images) — cas d'une surface non texturée et non-lambertienne avec une réflectance spéculaire uniforme ; l'illumination et les points de vue étant variables.

(malgré la présence de bruit haute fréquence) et estime particulièrement bien la réflectance.

Avec un cadre expérimental similaire (à l'exception qu'ici les coefficients diffus varient), la Figure 2.4 illustre une comparaison des résultats obtenus par l'algorithme que nous avons proposé avec ceux obtenus par l'algorithme de Pons, Keriven et Faugeras [67] qui est particulièrement robuste aux variations d'éclairage et aux spécularités grâce à l'utilisation de mesures de similarité telles que l'information mutuelle et la corrélation croisée. Comme le montre la figure, l'algorithme proposé dans [67] ne parvient pas à obtenir une forme correcte malgré la simplicité de la scène photographiée, tandis que notre méthode l'estime avec précision.

La Figure 2.5 montre enfin un exemple de résultats obtenus avec des images réelles d'une scène texturée et particulièrement non-lambertienne. L'illumination et le point de vue sont là encore différents pour chacune des images.

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 2 [53]

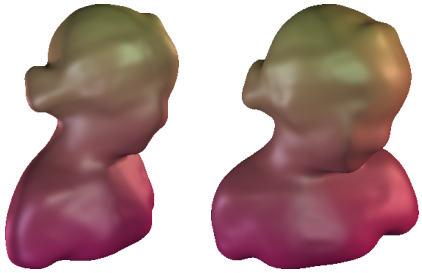
▷ PUBLICATIONS ASSOCIÉES : [53, 52, 54]

▷ COLLABORATEURS : Kuk-Jin Yoon (INRIA, équipe Perception) et Peter Sturm (INRIA, équipe Perception)

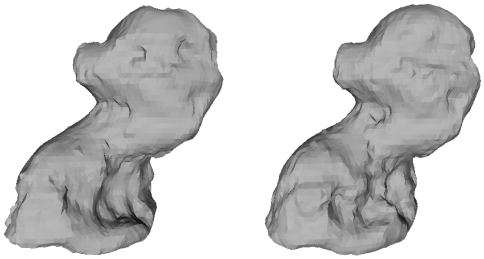
2.1.3 De la représentation "*level set*" aux maillages

Les travaux effectués par Pau et Kuk-Jin ont été développés dans le cadre continu via des représentations "*level set*". Malgré la souplesse indéniable que nous offre ces outils, nous nous sommes rendus compte pendant ces années de certaines limitations qui nous ont poussé à modéliser le problème directement avec des représentations discrètes. C'est pourquoi, avec Amaël Delaunoy, nous avons redéveloppé le calcul exact du gradient de l'erreur de reprojection (avec une fonctionnelle générique) dans le cas où la surface discrète est représentée par un maillage triangulaire.

De manière générale, une des limitations des gradients calculés dans le cadre continu est qu'en



(a) deux images d'entrée



(b) résultats en utilisant [67] (MI et CCL)

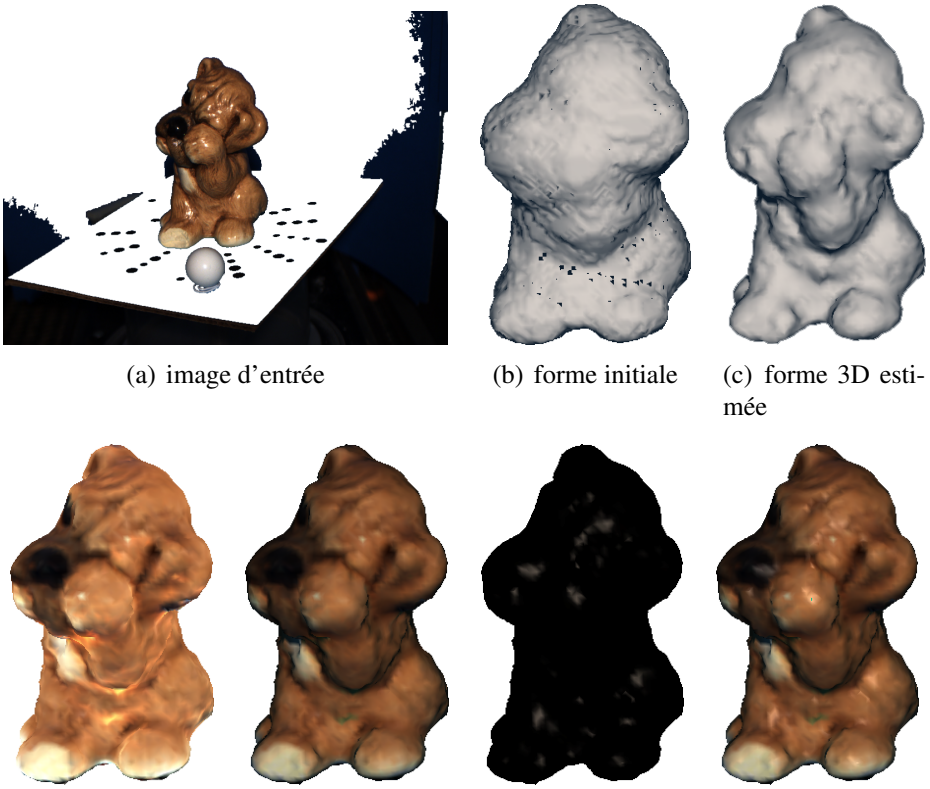
(c) résultat obtenu par notre algorithme

FIGURE 2.4 – Comparaison des résultats obtenus par l'algorithme que nous avons proposé avec ceux obtenus par l'algorithme de Pons, Keriven et Faugeras [67], avec un ensemble de 16 images de la "Bimba" — cas d'une surface non-lambertienne texturée avec une réflectance spéculaire uniforme ; l'illumination et les points de vue étant variants.

pratique, nous travaillons sur des représentations discrètes. Donc quelque soit la représentation utilisée, que ce soit une grille "*level sets*" ou un maillage, il devient nécessaire de calculer une discrétisation du gradient continu. Cependant, cette approximation introduit inévitablement des erreurs et amène des difficultés d'implémentation d'un point de vue numérique. Le travail effectué avec Amaël avec des maillages triangulaires permet de simplifier cet aspect et d'être plus rigoureux à ce niveau là.

Par ailleurs, dans le cas continu, des instabilités numériques dues à l'introduction des équations de radiance dans la fonction coût (instabilités déjà rencontrées dans les méthodes variationnelles en "*shape from shading*" et mises en évidence dans ce contexte par Soatto et ses collègues [61]) nécessitent la mise en oeuvre de certaines astuces comme l'introduction de champs auxiliaires approximant les normales à la surface et qui ont pour effet regrettable de régulariser considérablement la solution. Face à ce problème nous avions l'intuition que l'utilisation dès le départ de représentations discrètes nous permettrait d'éviter l'introduction de ces termes ; ce qui a été confirmé en pratique dans nos algorithmes exploitant la réflectance (les algorithmes équivalents qu'Amaël a implementés n'utilisent pas ces champs auxiliaires et les coefficients de régularisation qu'il a utilisés en pratique sont quasi nuls).

Enfin, une des difficultés pratiques que nous avions rencontrées avec Kuk-Jin était due à la faible résolution de notre grille "*level sets*" (due aux limites de capacité en mémoire de l'ordinateur). Si une meilleure gestion de la mémoire dans l'implémentation de la librairie "*level sets*" pouvait peut-être un peu améliorer les choses, le passage aux maillages nous a permis d'accéder très facilement à des résolutions significativement plus grandes. Ce qui, avec le point précédent, nous



(a) image d'entrée

(b) forme initiale

(c) forme 3D estimée

(d) image correspondant à la composante diffuse estimée ré-éclairée ("*shading*")

(e) image correspondant à la composante diffuse estimée

(f) image correspondant à la composante spéculaire estimée

(g) image resynthétisée

FIGURE 2.5 – Résultats pour un ensemble d'images réelles du "*saddog*" (59 images) — scène réelle texturée et particulièrement non-lambertienne. L'illumination et le point de vue sont différents pour chacune des images.

a permis d'obtenir des résultats beaucoup plus précis et plus stables que précédemment.

L'intérêt des outils développés par Amaël a été largement démontré dans divers cadres et applications, en particulier pour la régularisation de formes, en stéréovision et en intégration multi-vues de champs de normales.

La Figure 2.6 illustre les résultats que nous avons obtenus avec notre algorithme de stéréovision multi-vues minimisant correctement l'erreur de reprojection avec une surface représentée par un maillage triangulaire, sur la base de données des jeux d'essai "Dino" et "Temple" de Middlebury [68].

Attardons nous maintenant un peu sur l'intégration multi-vues de champs de normales. Ici, nous supposons que nous disposons de plusieurs images des normales à la surface vue de plusieurs points de vue. A chaque pixel de ces images est donc associé le vecteur normal à la surface au point de la scène *qui est vu à travers ce pixel*. De telles images de normales peuvent être obtenues par exemple via de la stéréovision photométrique. Comme dans la plupart des problèmes de vision, les images sont supposées ici être associées à une projection et ne contenir que l'information des points visibles. Le problème d'intégration multi-vues de ces images de normales peut alors s'écrire tout naturellement comme un problème d'optimisation de l'erreur de reprojection entre les normales du modèle estimé et les normales fournies par les données. Les Figures 2.7 et 2.8 montrent deux exemples de résultats d'intégration multi-vues de champs de normales avec des images synthétiques. Pour le test "Fandisk" (Fig. 2.7), le maillage original a deux fois plus de vertex et de triangles que le maillage reconstruit. Ici, grâce au terme de contour apparaissant dans le gradient de l'erreur de reprojection, les arêtes des triangles obtenus se placent et correspondent parfaitement bien aux contours présents dans les images ; ceci même si la résolution du maillage n'est pas très élevée. Ceci démontre que cette méthode est particulièrement pertinente pour reconstruire des surfaces contenant les arêtes particulièrement fines, ce qui à notre connaissance n'est pas le cas avec les méthodes "*level sets*". La Figure 2.8 montre quant à elle la capacité de l'algorithme à traiter des surfaces complexes. Plus d'exemples, en particulier des exemples avec des images réelles sont disponibles dans l'Article 3 de la partie II [5].

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 3 [5]

▷ PUBLICATIONS ASSOCIÉES : [5, 9, 8]

▷ COLLABORATEURS : Amaël Delaunoy (INRIA, équipe Perception), Peter Sturm (INRIA, équipe Perception), Pau Gargallo (INRIA, équipe Perception) et Jean-Philippe Pons (ENPC, CERTIS)

FIGURE 2.6 – Séquences d'images "Dino" et "Temple" [68]. De gauche à droite : une des seize images d'entrée ; images des fonds estimées (multipliées par deux pour la visualisation) ; radiances estimées ; maillages estimés vus d'un point de vue différent.

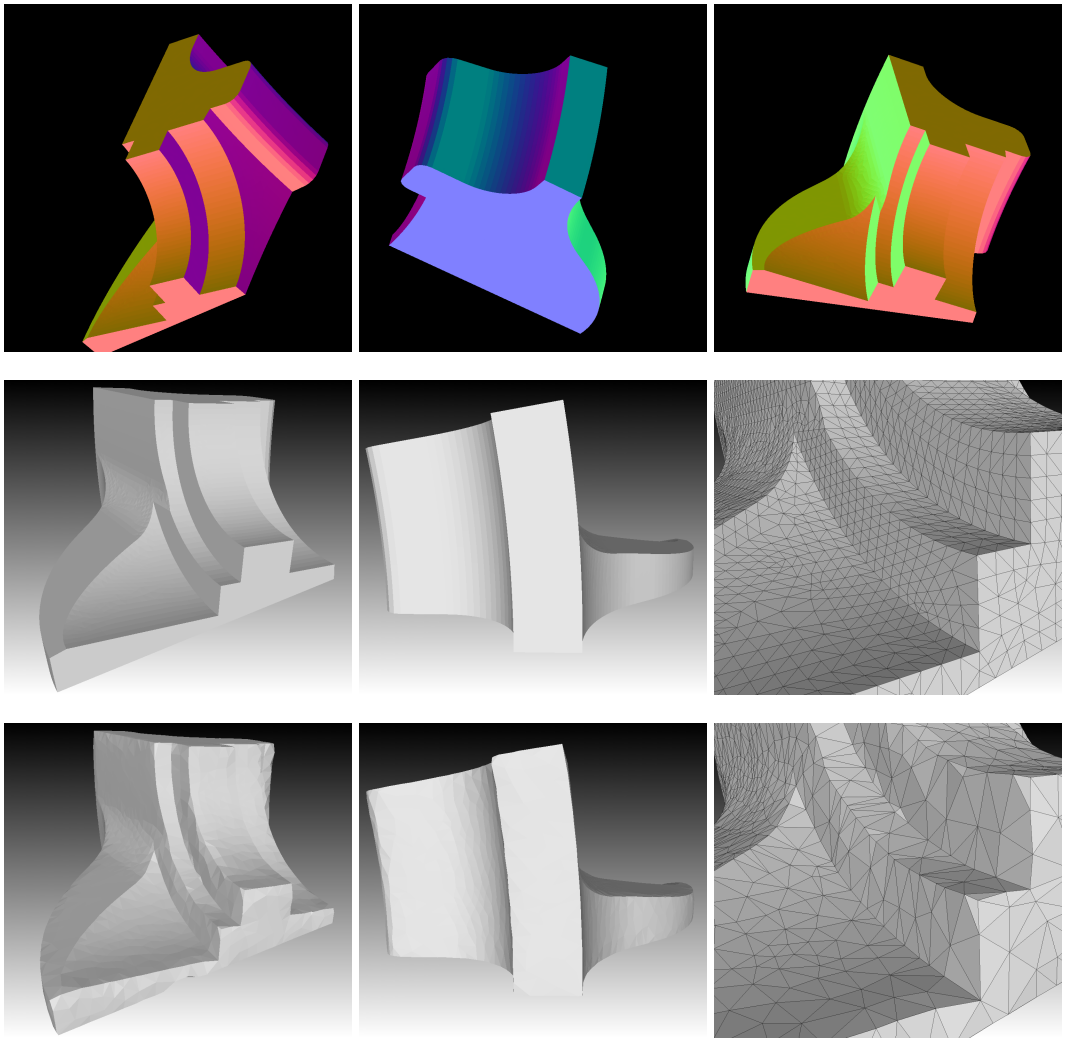


FIGURE 2.7 – Exemple “Fandisk”. De haut en bas : 3 des 24 images d’entrée montrant les cartes de normales de l’objet d’intérêt ; modèle CAD original (“*groundtruth*”) ; maillage reconstruit. La dernière colonne donne des détails du maillage estimé avec les triangles associés. Le terme de contour du gradient mis au jour par nos travaux permet aux arêtes du maillage de bien coller avec les discontinuités présentes dans les images qui sont générées par les auto-occultations ; ceci sans utiliser de termes additionnels de forçage du type “ballooning” [73] ou enveloppe visuelle [59, 69].

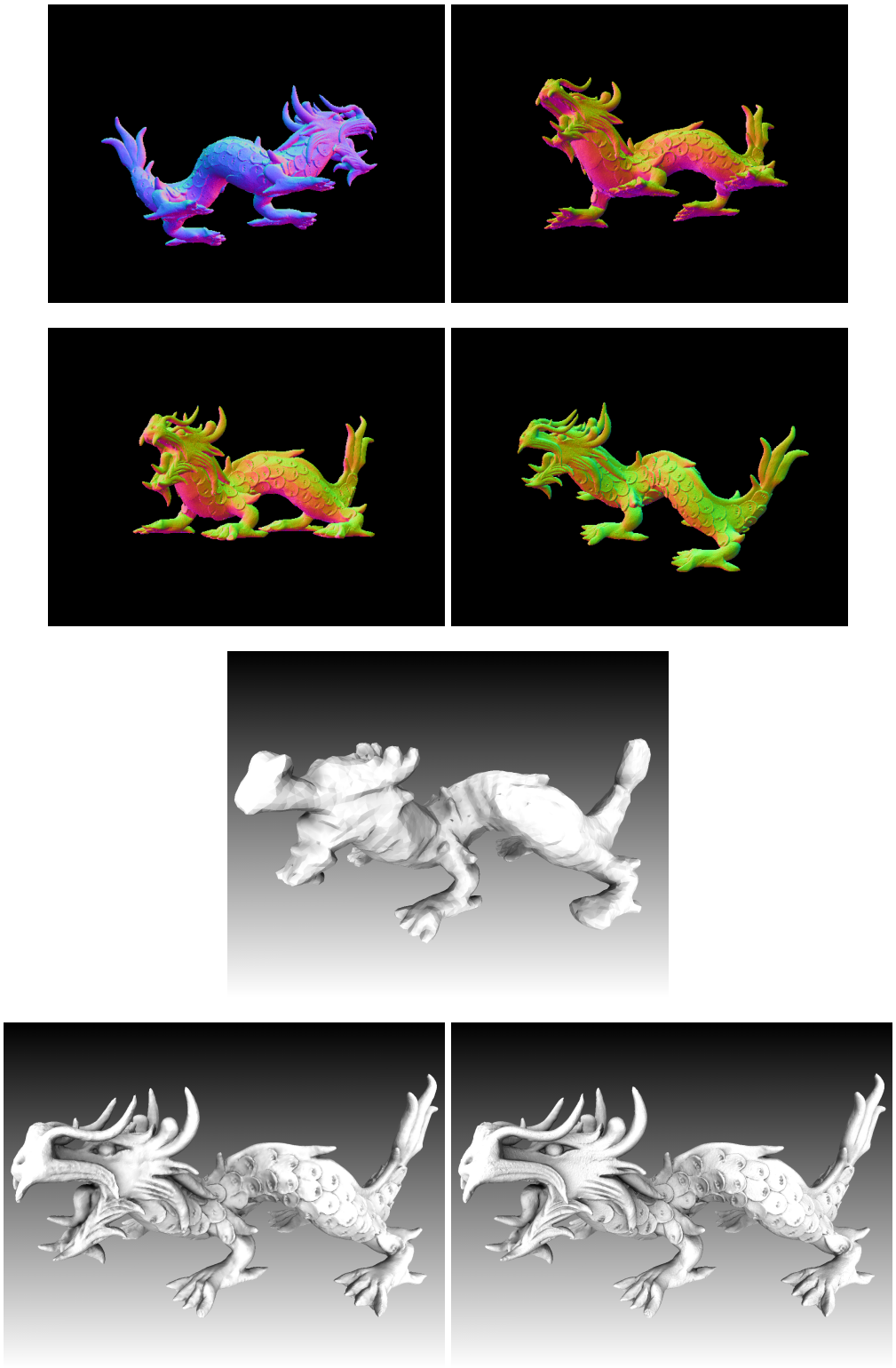


FIGURE 2.8 – Exemple du “Dragon” [55]. Deux premières lignes : 4 des 24 images d’entrée (640×480) montrant les cartes de normales de l’objet d’intérêt ; ligne du milieu : surface initiale ; Ligne du bas : surface reconstruite et surface d’origine “*ground truth*”. L’affichage des surfaces (“*rendering*”) est effectué en utilisant le “*flat shading*” de façon à pouvoir visualiser tous les détails.

2.1.4 Segmentation convexe sur des maillages

Revenons maintenant à la problématique même de la reconstruction multi-vues “3D + apparence”. L’algorithme développé par Kuk-Jin (dans sa version la plus générique) se place sous l’hypothèse que les propriétés de réflectance de l’objet pris en photographie peuvent varier spatialement. L’algorithme estime alors les propriétés de réflectance, plus exactement tous les paramètres du modèle de réflectance, en chacun des points de la surface. Cependant, dans de très nombreux cas, en particulier pour presque tous les objets et scènes créés par l’homme, ceux-ci ne sont constitués qu’avec très peu de matériaux. Il est alors particulièrement pertinent de segmenter la surface en plusieurs classes de matériaux homogènes pendant le processus de reconstruction 3D, comme cela l’a déjà été proposé par exemple dans [62, 63] dans le cadre des “*level sets*”. Nous n’avons alors plus qu’à estimer les paramètres de réflectance pour chacune des catégories de matériaux au lieu de devoir les estimer pour chacun des points de la surface. Par exemple, si l’objet est simplement constitué de deux matériaux, alors nous avons seulement besoin de récupérer deux ensembles de paramètres caractérisant les deux modèles de réflectance.

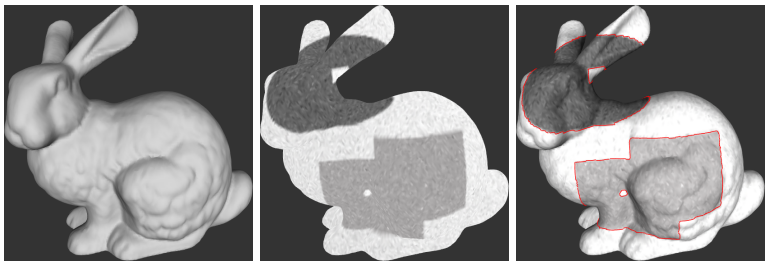
Avec Amaël, nous nous sommes alors intéressés au problème de la segmentation de données (de nature quelconque) définies sur une surface en un ensemble de régions distinctes. En nous appuyant sur les techniques les plus récentes et les plus performantes de segmentation d’images en plusieurs régions (segmentation variationnelle convexe basée sur la variation totale), nous avons proposé une méthode originale de segmentation [1] de données sur les variétés qui est robuste à l’initialisation et robuste au bruit. Nous avons montré comment implémenter la méthode lorsque la surface est représentée par un maillage triangulaire ; en particulier comment calculer le gradient de la variation totale dans ce cadre. Enfin, nous avons expliqué comment cette méthode pouvait être naturellement combinée avec les algorithmes de reconstruction 3D multi-vues estimant à la fois la forme 3D et la réflectance (via des modèles paramétriques).

Les Figures 2.9 et 2.10 montrent des exemples de notre algorithme sur le lapin de Stanford texturé avec des images de synthèse dans le cas binaire de deux régions constantes par morceaux. Du bruit gaussien a été ajouté sur la texture. Notre approche présente de bons résultats et montre la solution binaire finale. De plus nous voyons que la méthode est robuste à l’initialisation. Dans cet exemple, les contours actifs ou les *level sets* tomberaient dans le minimum local le plus proche de leur initialisation. Néanmoins, comme nous sommes en présence d’une méthode globale ici, segmenter une région particulière doit être fait en rajoutant des contraintes additionnelles.

La Figure 2.11 montre les résultats de la segmentation avec des données fortement bruitées avec l’approche proposée et l’algorithme K-moyennes. Notre approche qui, comparée à K-means, utilise une régularisation permettant une cohérence spatiale de l’information, propose une segmentation proche de celle souhaitée et non bruitée. En plus d’être robuste à l’initialisation, notre approche est également robuste au bruit. Nous avons ensuite testé l’approche multi-régions sur les données provenant de [65], voir Figure 2.12.

Finalement, la Figure 2.13 montre un exemple avec un maillage 3D obtenu par reconstruction 3D utilisant [70, 75]. La figure montre la segmentation de la radiance de la surface en trois régions distinctes. Nous retrouvons parfaitement les régions de la peau, du pantalon et du t-shirt.

1. ...segmentation en plusieurs régions.



(a) (b) (c)

FIGURE 2.9 – Résultat de la segmentation sur les données synthétiques du lapin de Stanford. (a) Forme d'entrée. (b) Surface d'entrée avec une texture de synthèse. (c) Maillage texturé et contour final retrouvé par notre approche.

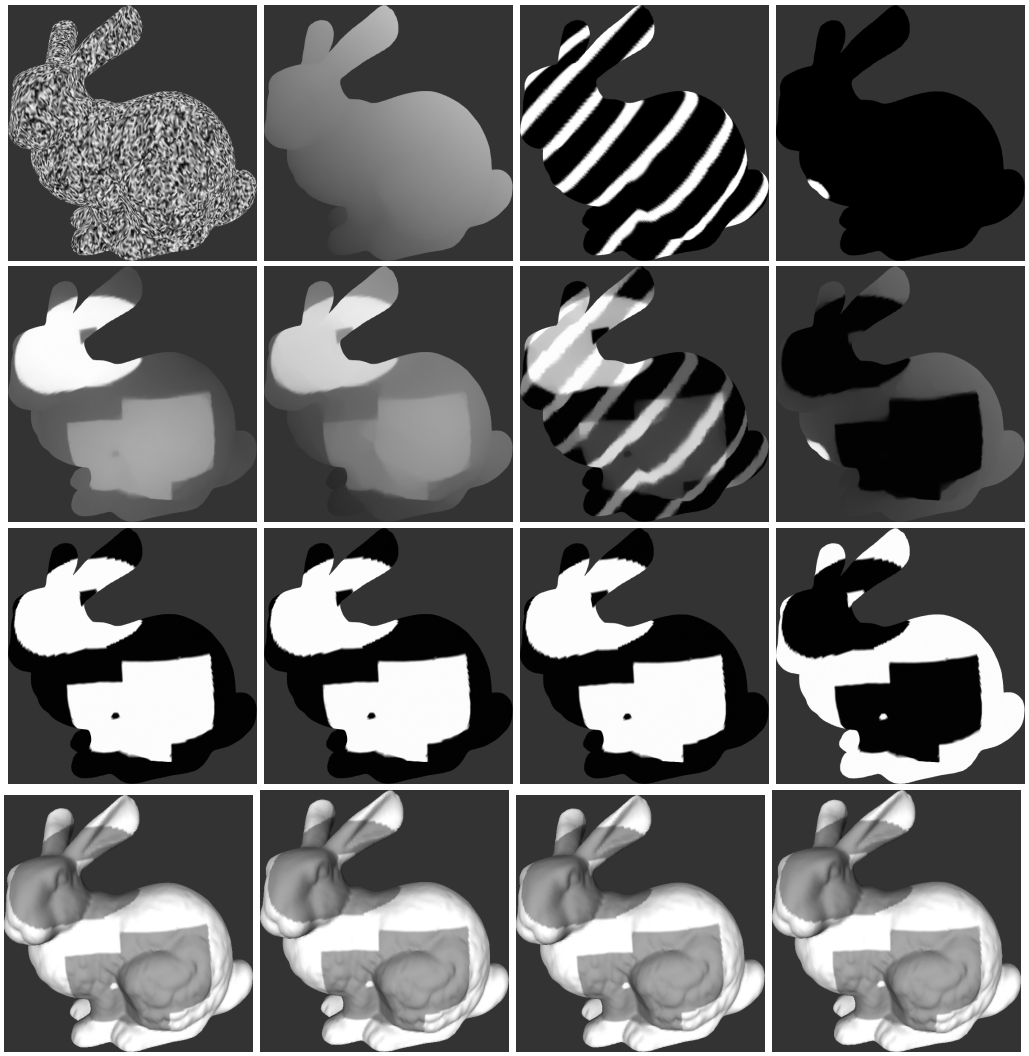
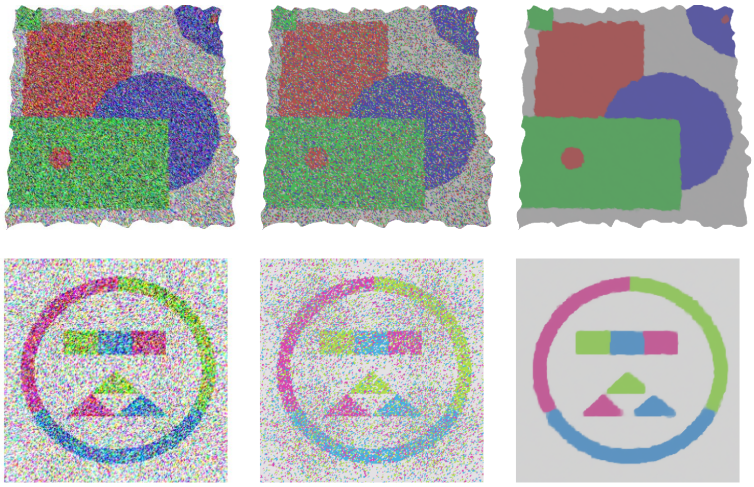


FIGURE 2.10 – Évolution sur les images de synthèse du lapin. Différentes initialisations (première ligne) ; Valeurs intermédiaires (seconde ligne) ; Solutions obtenues (troisième ligne) ; Valeurs moyennes obtenues (quatrième ligne) avec ombrage.



(a) (b) (c)

FIGURE 2.11 – Résultats de segmentation sur maillage dans le cas multi-régions pour différents exemples de synthèse inspirés de [64]. (a) Maillage texturé d'entrée. (b) Segmentation par l'algorithme K-moyennes. (c) Valeurs moyennes de chaque région obtenues par notre approche.

(a) (b) (c)

FIGURE 2.12 – Résultats de segmentation sur les maillages dans le cas multi-régions. Ligne du haut : données du cheval et sa segmentation en trois régions. Ligne du bas : Labélisation en quatre régions des données du papillon. (a) Maillage texturé d'entrée ; (b) Valeurs moyennes estimées pour chaque classe ; (c) Une des régions segmentées.

FIGURE 2.13 – Résultat de segmentation sur un maillage obtenu par stéréovision multi-vues. Vue de face (ligne du haut) et de dos (bas). Maillage coloré original et la forme 3D associée (gauche). Résultat de la segmentation en trois régions par notre approche (milieu). Valeurs moyennes pour chaque région segmentée (droite).

Malgré ces excellents résultats, nous n'avons pas approfondi d'avantage cette question (bien que de nombreuses améliorations et applications significatives pouvaient être développées) car nous voulions absolument travailler sur la question de l'utilisation des modèles paramétriques dont nous voulions nous affranchir.

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 4 [4]

▷ PUBLICATIONS ASSOCIÉES : [4, 7]

▷ COLLABORATEURS : Amaël Delaunoy (INRIA, équipe Perception), Ketut Fundana (université de Malmö) et Anders Heyden (université de Lund)

2.1.5 S'affranchir des modèles paramétriques tout en exploitant les propriétés de la réflectance

De mon point de vue, la principale mais essentielle limitation des méthodes développées avec Kuk-Jin et Amaël (qui estiment à la fois la réflectance et la forme 3D) est qu'elles nécessitent de connaître et de choisir *a priori* un modèle paramétrique de réflectance correspondant aux matériaux constituant la scène. Cette limitation pourrait être réduite en faisant en plus de la sélection automatique de modèles et/ou en utilisant plusieurs modèles combinés avec de la segmentation de la surface (comme présentée dans la section précédente). Mais cela serait certainement très lourd à mettre en œuvre et peut-être peu efficace, d'autant plus qu'en pratique, même le meilleur des modèles paramétriques considérés ne sera inévitablement qu'une approximation (parfois très grossière) de la réalité ; ce qui est d'ailleurs une autre limitation intrinsèque de ces méthodes basées sur des modèles paramétriques.

Comment faire alors pour arriver à prendre en compte et à exploiter des propriétés de la réflectance dans le but de gagner en précision et en robustesse (en obtenant de l'information directement sur les normales en plus de l'information de profondeur fournie par la correspondance) sans utiliser de modèles paramétriques ?

Malheureusement, la littérature ne nous apporte que très peu de pistes pour répondre à cette question. Je ne citerai ici que les deux références qui sont particulièrement remarquables : les travaux de Hertzmann et Seitz développant des algorithmes de reconstruction 3D basés sur des exemples [72, 60] et les méthodes de stéréovision d'Helmholtz développées par Zickler et ses collègues [77, 76].

Cette dernière approche (la stéréovision d'Helmholtz) nous semble beaucoup moins contraignante et plus prometteuse que la première. Cette approche consiste en fait à tirer partie du principe de réciprocité de Helmholtz qui est exploitée en prenant une paire d'images avec une unique source de lumière éclairant la scène, où les positions du centre de la caméra et de la lumière sont interchangées. Dans cette configuration particulière, les deux radiances d'un même point de la surface reçues dans chacune des deux images sont liées par une relation indépendante de la réflectance de la surface en ce point. Cette relation peut ensuite être utilisée pour estimer (ou au moins contraindre) la normale à la surface en ce point. Ainsi, à ma connaissance, ces méthodes sont les seules permettant d'estimer avec précision les normales à la surface indépendamment de la réflectance de celle-ci. Aucune connaissance de la BRDF, aucun exemple d'images ("*samples*") et aucun invariant photométrique ne sont ici nécessaires.

Malgrés leur potentiel énorme, les méthodes de stéréovision d'Helmholtz étaient cependant limitées à la stéréovision binoculaire ou à la reconstruction de cartes de profondeur. Avec Amaël, il nous a paru alors essentiel d'étendre ces méthodes pour permettre la reconstruction de la forme 3D complète des objets d'une scène. Ce que nous avons fait. Ainsi, nous sommes désormais capables de reconstruire la forme 3D complète d'objets (opaques) ayant une BRDF arbitraire et inconnue.

Pour arriver à ce résultat, contrairement aux méthodes précédentes qui étaient basées image et ne s'appuyaient que sur une carte de profondeur, nous avons implémenté une méthode basée objet utilisant un modèle déformable (maillage triangulaire) représentant la surface 3D complète de la

scène. Cette représentation basée objet permet par ailleurs de gérer plus facilement les occlusions (“*self occlusions*”) et les ombres portées au cours du processus d’optimisation. Elle permet aussi d’exploiter naturellement les relations entre la géométrie d’un point de la scène et la normale à la surface (relations qui étaient mal exploitées dans les travaux précédents) ; exploitation qui est encore plus facilitée via l’utilisation des maillages triangulaires.

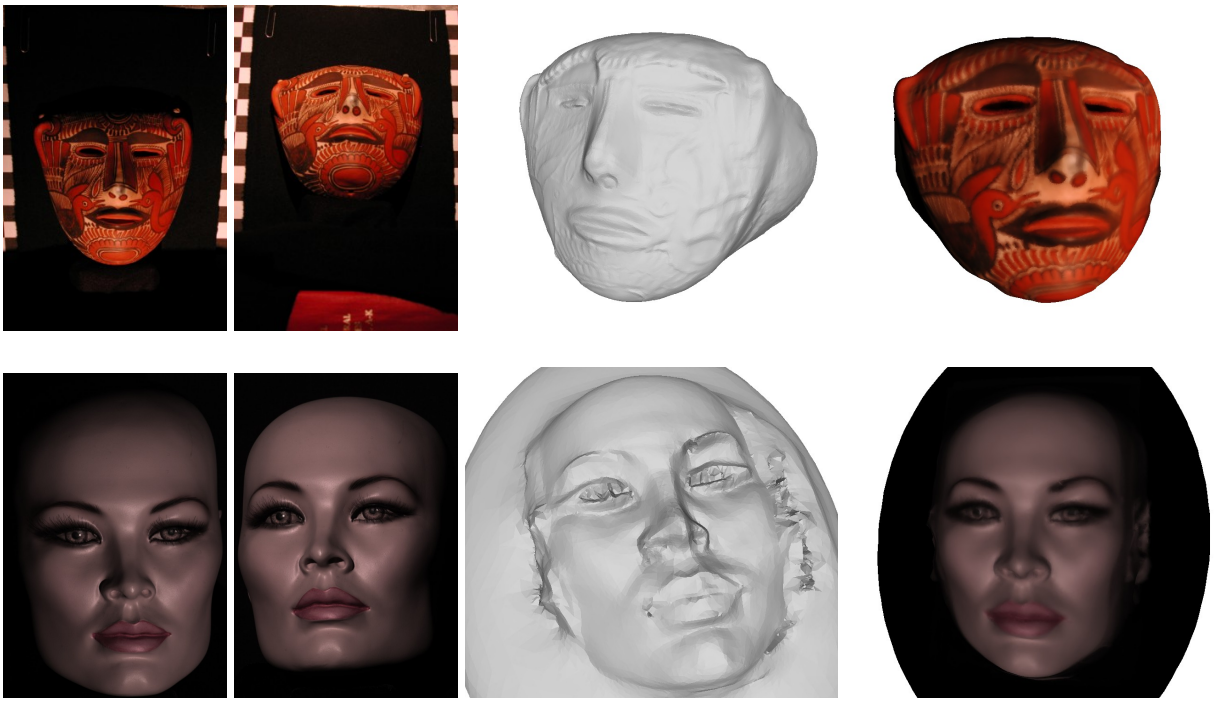


FIGURE 2.14 – Base des données “Mannequin” (8 paires d’images réciproques) et “Masque” (18 paires d’images réciproques) : résultats obtenus par la méthode proposée avec ces deux bases de données contenant des apparences complexes et variant spatialement. De gauche à droite : une des 18 paires caméra / lumière (respectivement 8) ; résultat final ; maillage final texturé par la moyenne des couleurs reprojetées.

La Figure 2.14 montre les résultats obtenus avec les bases de données classiques du “Masque” et du “Mannequin” [77, 78]. La Figure 2.15 montre les résultats obtenus avec de nouvelles bases de données que nous avons générées afin de reconstruire la forme 3D complète des scènes. Dans tous les cas (Figures 2.14 et 2.15), les images sont des images réelles de scènes non lambertiennes dont nous n’avons aucune idée des propriétés de réflectance (et nous n’essayons pas de les connaître !) ; propriétés qui varient d’ailleurs spatialement.

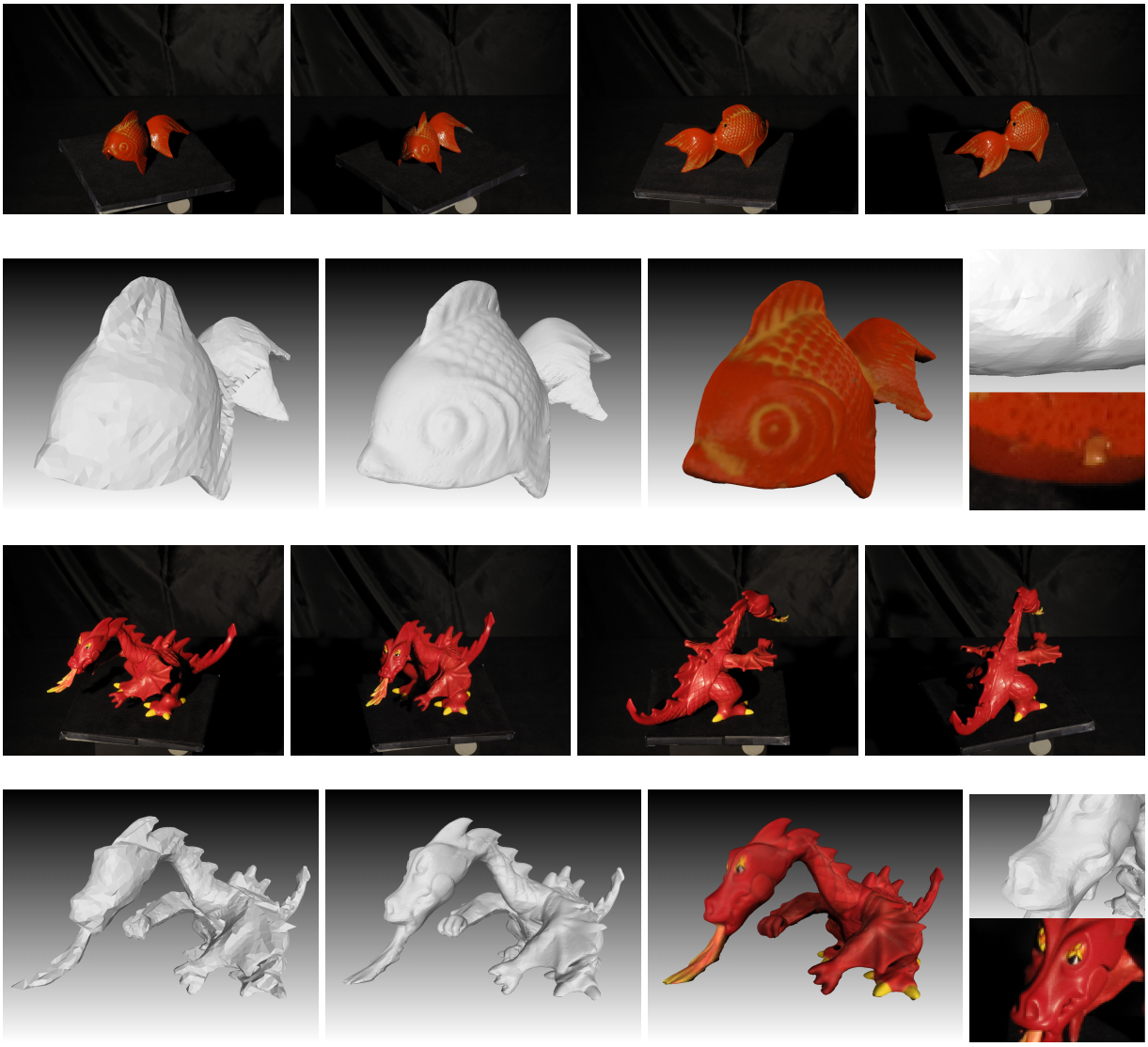


FIGURE 2.15 – Base de données “Poisson” et “Dragon” (18 paires d’images réciproques 1104×828) : résultats obtenus par la méthode proposée avec deux jeux de données réelles complètement 3D. Première et troisième lignes : deux des 36 paires caméra / lumière . Deuxième et dernière lignes : enveloppe visuelle utilisée pour l’initialisation ; maillage 3D reconstruit ; maillage reconstruit texturé avec la moyenne des couleurs rétro-projetées à partir des caméras ; zoom sur une zone du maillage reconstruit et sur une image d’entrée correspondante.

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 5 [6]

▷ PUBLICATIONS ASSOCIÉES : [6]

▷ COLLABORATEURS : Amaël Delaunoy (INRIA, équipe Perception) et Peter Belhumeur (université de Columbia)

2.1.6 Retour aux sources : Le "*Shape From Shading*"

Pendant mes trois années de thèse je me suis totalement consacré à ce fameux problème qu'est le *shape from shading*. J'étais véritablement passionné (voire obsédé) par les questions qu'il pose. Du coup, bien que n'y travaillant plus vraiment, j'avoue avoir toujours gardé ce truc là dans un coin de ma tête. Les invitations incessantes de relectures d'articles sur ce sujet ne m'ont d'ailleurs pas aidé à faire mon deuil. ⌣̈

Avec le recul, ma vision des choses a un petit peu mûrie ; tout d'abord, concernant l'intérêt même de travailler sur cette question. Bien que ce problème soit passionnant car très difficile et assez énigmatique, pendant longtemps je n'ai pas vraiment réussi à me faire une idée sur son véritable intérêt. Pourquoi se restreindre à une image alors que dans quasiment toutes les applications, nous pourrions en disposer de plusieurs [2] ? A en voir le peu de motivations données en introduction des articles de vision par ordinateur sur le sujet [3] et le faible intérêt que ce sujet suscite, mon sentiment est que la communauté ne semble pas vraiment être convaincue non plus. En fait, aujourd'hui je pense que le problème du "*shape from shading*" est intéressant et important (seulement) car c'est un problème jouet ; un problème jouet qui est épuré et qui offre un cadre expérimental idéal pour mieux comprendre l'information brute présente dans l'ombrage (présent lui même dans les images). Dès que nous disposons de plusieurs images, cette information de base est alors complétée par d'autres informations, et est donc pour ainsi dire "noyée". Par exemple, dans le cas de la stéréovision et du "*shape from shading*" multivues, celle-ci est couplée voire effacée par l'information de correspondance ; ce qui rend les analyses beaucoup trop difficiles. Dans le cas de la stéréovision photométrique, l'information n'est plus "brute" en ce sens que nous analysons le croisement de plusieurs informations d'ombrage... L'analyse du problème de "*shape from shading*" permet donc de fournir un ensemble de clefs permettant de comprendre *si*, *quand* et *combien* l'ombrage contient de l'information exploitable en reconstruction 3D. Mais l'intérêt de ce problème est plus théorique qu'algorithmique : développer des algorithmes de "*shape from shading*" n'est pas une finalité car en pratique, dans les applications, nous pouvons disposer de plusieurs images, et dans ces cadres nous n'utiliserons pas *a priori* les mêmes techniques algorithmiques. Par contre, le développement d'algorithmes de "*shape from shading*" est essentiel pour tester expérimentalement la théorie de façon à pourvoir la valider ou l'infirmer. C'est pourquoi ceux-ci doivent être particulièrement rigoureux et fiables ! Par exemple, les algorithmes peuvent

2. ...même en microscopie électronique, on fait aujourd'hui de la stéréo !

3. ...les articles se limitent trop souvent à des justifications du style "Le *shape from shading* a une longue tradition en vision par ordinateur", ou "Le *shape from shading* est un problème majeur en vision par ordinateur" ou encore "alors que la reconstruction 3D à partir de plusieurs images est relativement mature, la reconstruction à partir d'une seule image reste toujours un problème difficile"...

permettre de voir expérimentalement les limites des modèles et de l'analyse proposés. Ils permettent de tester les conditions aux limites, la stabilité et la robustesse au bruit, aux surbrillances etc.

J'ai aussi compris assez tardivement l'importance de la question "pourquoi la notion de solutions de viscosité (sur laquelle je me suis appuyé dans mes travaux de thèse) apporte ici une solution plutôt satisfaisante ?" ; question que plusieurs personnes (en particulier des mathématiciens comme Pierre Bernhard) m'avaient posée une fois ma thèse terminée. La relecture de nombreux articles qui ont suivi en SFS (nécessairement critiques sur mon travail) m'ont aussi donné envie de revenir et de réfléchir sur cette question. J'ai très rapidement trouvé la réponse que j'ai eu l'occasion de rédiger lorsque Daniel Cremers m'a invité à écrire un article pour l'"*Encyclopedia of Computer Vision*" sur les solutions de viscosité [3].

La dernière chose qui m'a poussée à revenir dernièrement sur ce problème est liée au fait que la communauté n'a vraisemblablement pas complètement cerné où est la véritable contribution dans mon travail de thèse. Tous les commentaires que j'ai pu voir par la suite (via les diverses citations de mes articles) se focalisent sur le modèle de caméra en perspective, alors que de mon point de vue, cet aspect n'est finalement qu'accessoire. Le point clé est en fait l'éclairage. Aussi, ma principale motivation dans les quelques travaux que j'ai menés avec Nitin Jindal et Alejandro Deymonnaz, c'était d'enfoncer le clou ; c'est à dire de mettre particulièrement en évidence ce point clé. Ces derniers travaux nous permettent en particulier de démontrer que le modèle de caméra n'est pas un élément critique ; il permet simplement de simplifier l'analyse. Par ailleurs, nous avons trouvé un autre type d'éclairage (différent de celui que nous présentons dans [36] qui repose sur une source de lumière ponctuelle à proximité de l'objet) qui permet de mieux poser le problème qu'avec le type d'éclairage utilisé classiquement (une unique source de lumière ponctuelle localisée à l'infini). Cet autre type d'éclairage est basé sur un éclairage diffus. Malheureusement, seuls quelques prémices de ce travail ont été publiés [43]. Nous rédigeons actuellement un article détaillant ces derniers travaux...

▷ ARTICLE REPRÉSENTATIF DISPONIBLE DANS LA PARTIE II : Article 6 [43]

▷ PUBLICATIONS ASSOCIÉES : [43, 3]

▷ COLLABORATEURS : Nitin Jindal (INRIA, équipe Perception), Alejandro Deymonnaz (INRIA, équipe STEEP), Fabio Camilli (Université de Rome "La Sapienza") et Stefano Soatto (UCLA)

2.2 Brève description du projet scientifique de l'équipe STEEP

Très rapidement, dès 2OO8, j'ai eu envie de me lancer dans une nouvelle aventure et de travailler sur des questions plus en accord avec mes convictions et priorités, autour des problématiques soulevées par le développement durable. Dès le départ, il m'a paru essentiel d'arriver à constituer une équipe pluridisciplinaire. Ceci a abouti en janvier 2010 à la création de STEEP, Action Exploratoire INRIA et équipe du centre Grenoble-Rhône-Alpes. STEEP est initialement constituée d'Elise Arnaud (chercheuse en informatique et mathématiques appliquées), Pierre-Yves Longaretti (physicien), François Mancebo (géographe-urbaniste) et moi même (chercheur en informatique et mathématiques appliquées). Récemment Peter Sturm (informaticien) nous a rejoint. L'équipe compte aussi aujourd'hui deux post-doctorants et un doctorant. STEEP a pour objectif de développer des outils numériques d'aide à la décision permettant de faciliter la mise en oeuvre de politique de développement durable à l'échelle locale. Notre ligne de conduite s'appuie sur la conviction que, vu l'intrication et la complexité des problèmes, seules des approches intégrées prenant en compte toutes les dimensions des problèmes peuvent apporter des solutions vraiment soutenables. C'est pourquoi, nous travaillons en particulier sur la modélisation des interactions entre environnement, économie et société. Notre article [20] correspondant à l'article 7 dans le recueil ci-après, détaille nos motivations et choix d'échelle via quelques exemples concrets.

En pratique, les travaux de STEEP suivent quatre axes de recherche fortement interconnectés :

2.2.1 Développement de modèles intégrés systémiques prenant en compte les interactions entre environnement, économie et société

Le problème que nous considérons est intrinsèquement pluridisciplinaire : il fait appel aux sciences humaines, à l'écologie ou encore aux sciences de la planète. La modélisation des phénomènes mis en jeu doit rendre compte d'un grand nombre de facteurs de natures différentes, interagissant par des relations fonctionnelles variées. Ces dynamiques hétérogènes sont *a priori* non linéaires, et complexes : elles peuvent présenter des mécanismes de saturation, des effets de seuils ou encore être densité dépendantes. Les difficultés sont amplifiées par les fortes interconnexions du système (présence de boucles de rétroactions importantes), et par la spatialité des interactions qui sont multi-échelles. Les phénomènes environnementaux et sociaux sont en effet guidés par la géométrie des territoires sur lesquels ils apparaissent. Le climat ou l'urbanisation en sont des exemples typiques. Ces processus spatiaux font intervenir des relations de proximité ou de voisinage, comme entre deux parcelles de sol voisines, ou des relations à des niveaux macroscopiques entre niveaux d'organisation. L'aspect multi-échelle découle de la prise en compte simultanée dans la modélisation d'acteurs de nature différente et évoluant à des échelles (spatiales et temporelles) spécifiques. Par exemple, l'échelle à laquelle nous devons considérer l'évolution de la ruralité est très probablement différente de celle à laquelle nous modéliserons les phénomènes biologiques permettant de traiter convenablement les problèmes de biodiversité.

Dans ce contexte, le développement de modèles intégrés systémiques souples (évolutifs, modulaires, ...), performants, réalistes et simples d'utilisation (codeurs, modélisateurs, et utilisateurs finaux) est un challenge en tant que tel. Quelles représentations mathématiques et quels outils informatiques utiliser ? Aujourd'hui de nombreux outils sont utilisés : par exemple les automates cellulaires (utilisés par exemple dans le modèle LEAM [71]), les modèles d'agents (ex. URBAN-

SIM [57]), la dynamique des systèmes (ex. World3 [66]), des gros systèmes d'équations ordinaires (ex. modèles d'équilibre tels que TRANUS [56]) . . . Chacun de ces outils ayant tous des forces et des faiblesses, est il nécessaire d'inventer d'autres représentations ? Quel niveau de modularité est il pertinent ? Comment arriver à avoir des modèles très modulaires tout en gardant un ensemble très cohérent et facile à calibrer ? Est il préférable d'utiliser le même outil pour tout le système ou pouvons nous librement changer de représentations pour chaque sous système considéré ? Comment facilement et bien gérer les différentes échelles ? (difficulté apparaissant en particulier lors de la calibration des modèles). Comment obtenir des modèles s'adaptant automatiquement à la granularité des données et qui sont toujours numériquement stables ? (lien aussi avec la calibration et la propagation des incertitudes). Comment élaborer des modèles qui puissent être calibrés avec un effort raisonnable, compatibles avec les ressources (humaines et matérielles) des agences et sociétés de conseils qui les utilisent ?

2.2.2 Calibration semi-automatique des modèles intégrés

Lors de l'application de tels modèles intégrés, une des difficultés principales réside dans les processus de calibration. Il ressort en effet des retours d'expériences que cette tache est particulièrement délicate et nécessite une importante expertise expérimentale. En pratique, les modèles d'équilibre, tels que TRANUS par exemple, convergent difficilement et les algorithmes actuels n'apportent pas les moyens permettant à l'utilisateur de corriger les paramètres de façon à obtenir la convergence. Or, cette convergence est une étape clé de la calibration : il s'agit de reproduire de manière stable un état de référence.

D'autre part, calibrer ce type de modèles nécessite l'estimation d'un très grand nombre de paramètres qui sont difficiles à obtenir à partir de données et qui sont plus ou moins bien évalués en pratique. En général, c'est en particulier le cas pour le modèle TRANUS, ces paramètres sont aujourd'hui ajustés à la main via un processus long d'essais/erreurs ; la calibration prend typiquement 6 mois pour des modèles de taille moyenne (une centaine de zones géographiques pour quelques dizaines de secteurs comprenant les secteurs économiques, les différentes catégories de la population, etc.). Actuellement, il n'existe pas de moyen d'optimiser ces paramètres de manière automatique ou semi-automatique de façon à s'assurer que la réalité de référence observée est correctement modélisée. Il s'agit ici non plus seulement de converger vers un état stable mais de s'assurer que cet état correspond bien à la situation de référence choisie du système urbain. C'est une condition cruciale à la qualité des résultats prospectifs produits ensuite par le modèle.

Enfin, notons que pour obtenir des résultats robustes et fiables, il faut absolument prendre en compte les incertitudes présentes dans les données lors de cette dernière étape de calibration. Par exemple, ceci peut être fait de manière similaire à ce qui est fait en assimilation de données. Par ailleurs, lorsque des données sont aberrantes ("*outlier*"), celles-ci devraient être corrigées ou au moins détectées automatiquement. Notons enfin que des analyses de sensibilité peuvent fournir des outils particulièrement utiles pour réduire la dimension de l'espace des paramètres à optimiser et ainsi simplifier significativement les futures étapes de calibration.

2.2.3 Prise en compte des incertitudes et analyse de sensibilité

Les incertitudes apparaissent à plusieurs niveaux : elles vont de l'imprécision des données jusqu'aux incertitudes liées à l'absence de connaissance scientifique sur certains des processus concernés. Jusqu'à présent cette difficulté n'a quasiment pas été considérée dans ces domaines (modèles intégrés socio-économiques et environnementaux). Cependant, la prise en compte des incertitudes nous semble être un point essentiel pour plusieurs raisons.

Tout d'abord, il est fondamental de bien comprendre la propagation des erreurs et leurs impacts sur les résultats obtenus. Dans ce but, il est nécessaire de réaliser des analyses poussées sur les sources d'incertitudes et sur la sensibilité des différentes variables du modèle ou indicateurs [4] par rapport aux erreurs présentes dans les données d'entrée et les paramètres estimés. Cette analyse doit permettre de s'assurer que les valeurs des variables de sortie ou des indicateurs sont significatives par rapport aux incertitudes. En d'autres termes, l'objectif ici est donc de s'assurer de la robustesse des conclusions tirées par les analystes [5] par rapport aux incertitudes. Or à notre connaissance, à quelques exceptions près, les travaux de modélisation existant se limitent à donner des résultats prospectifs déterministes, sans évaluer la confiance que l'on peut avoir en ces résultats. Notons au passage, qu'il est important de distinguer au moins deux sources d'erreurs qui au final sont combinées dans les résultats. Il y a les erreurs dans les données dont l'importance sur le résultat final est due à des aspects structurels du modèle (aspects liés à des notions de stabilité et de robustesse). Et il y a les erreurs introduites par le modèle lui même dues au fait qu'il n'est qu'une approximation de la réalité. Une analyse de ces dernières est tout aussi fondamentale que la précédente et touche au problème de validation de modèles. Sur ce point, la littérature est extrêmement pauvre. Seuls peu d'articles proposent quelques pistes et suggèrent des embryons de méthodologies.

Si une analyse de sensibilité *a posteriori* permet de voir s'il est possible de tirer des conclusions significatives à partir d'un modèle implémenté, l'intérêt de ce type d'analyses ne se limite pas là. Une telle analyse sur les paramètres permet aussi de déterminer les principaux "*drivers*" du modèle, c'est à dire les paramètres déterminants : ceux qui ont un véritable impact sur la dynamique du modèle. L'identification de ces "*drivers*" est très importante pour les décideurs qui sont justement à la recherche de leviers d'actions (étape de prospective). Ensuite, des analyses de sensibilité *a priori* sont aussi très utiles pour avoir à l'avance une idée du degré de précision nécessaire dans les données et paramètres pour garantir la pertinence du modèle. Cette connaissance *a priori* est importante pour le technicien qui doit implémenter le modèle en pratique. Cela lui permet en effet de ne pas gaspiller inutilement de l'énergie dans l'élaboration trop précise de données qui n'ont finalement que peu d'influence. A l'inverse, cela lui permet de concentrer ses efforts sur les données sensibles. Enfin, comme mentionné précédemment, les analyses de sensibilité constituent des outils précieux permettant de réduire la dimension de l'espace des paramètres à optimiser dans la phase de calibration du modèle (en ne gardant que les paramètres fortement influents) ; ce qui en réduit significativement la difficulté.

Indépendamment de tout ceci, il est judicieux de prendre en compte les incertitudes présentes dans les données à l'intérieur même de l'algorithme de calibration du modèle. Nous proposons de faire ceci de manière similaire à ce qui est fait en assimilation de données. En encapsulant ces erreurs

4. Un indicateur est une information construite a posteriori à partir des variables de sortie du modèle. Cette information peut être plus ou moins quantitative. Elle peut être représentée par des valeurs mais cela peut aussi être des tendances…

5. ...qui utilisent les variables de sortie ou les indicateurs comme briques de base à leurs propres analyses.

dans l'algorithme de calibration lui même, nous limitons ainsi leur propagation. De même, il est important d'arriver automatiquement à détecter et éventuellement corriger les données aberrantes lors du processus de calibration.

Enfin, il est fondamental de remarquer ici une spécificité des modèles socio-économiques et environnementaux (modèles que nous considérons dans STEEP). Cette spécificité repose sur la présence inévitable de scénarios. Dans certains cas, ces scénarios relèvent du contrôle (au sens mathématique du terme) ou de très grandes incertitudes. Par exemple, il est commun d'utiliser des scénarios correspondant à des choix politiques dont il faut mesurer l'impact. Dans ce cas, ces scénarios relèvent du contrôle. Par ailleurs de nos jours, l'introduction de scénarios climatiques est quasi systématique. Ce deuxième type de scénarios relève de l'incertitude. Insistons sur le fait que ce genre d'incertitudes très importantes n'est pas présent dans les modèles physiques et géophysiques. Elles sont propres aux modèles socio-économiques et, à notre connaissance, elles n'ont jamais été étudiées formellement. Ces incertitudes ne peuvent pas être traitées de la même manière que le bruit présent dans les données ou que les erreurs dues aux approximations du modèle. Par ailleurs, pour avoir du sens, les analyses de sensibilité sur les données et paramètres, ainsi que les méthodes de calibration doivent être relativement robustes à ces incertitudes. Il nous paraît désormais essentiel de travailler aussi sur ces questions.

2.2.4 Apporter des éléments de réponse aux questions soulevées par le développement durable

Arriver à développer des modèles intégrés systémiques qui sont numériquement stables, qui sont robustes, que nous pouvons facilement calibrer et adapter et pour lesquels nous pouvons évaluer la confiance est crucial. Cependant, cela n'est pas suffisant. Notre objectif premier est d'apporter des éléments de réponses permettant la mise en oeuvre de politique de développement durable à des échelles locales. Donc nos modèles doivent être spécialement conçus pour aider à comprendre, analyser et en partie répondre aux problématiques auxquelles se heurte le développement durable. Nos modèles doivent donc prendre en compte les interactions clef permettant de répondre aux bonnes questions (et même peut-être de comprendre quelles sont ces bonnes questions !).

Parallèlement aux activités décrites ci-dessus, il me paraît alors crucial que nous travaillons conjointement avec nos partenaires des autres disciplines (biologie, géophysique, sciences économiques, politiques et sociales, etc.) pour faire avancer la réflexion autour de la soutenabilité, pour identifier les verrous et points de blocage puis enfin, pour arriver à proposer des leviers d'actions. Je pense sincèrement que les outils de modélisation sont des éléments clef et pourraient même être la moelle épinière dans un cadre pluridisciplinaire comme celui-ci, car vecteur d'échanges entre les disciplines. La modélisation nécessite une certaine rigueur et une formalisation qui facilitent en effet la communication et la compréhension mutuelle.

Par ailleurs, en plus d'arriver à avoir des modèles scientifiquement pertinents prenant en compte tous les bons ingrédients et aidant à progresser sur ces questions, un de nos objectifs est d'aller jusqu'à l'aide à la décision. Il faudrait donc aussi que nos modèles s'interfacent ou s'intègrent facilement avec des techniques telles que l'aide à la décision multicritère et qu'on développe des outils répondant aux besoins des agences. Nos modèles doivent donc entre autre retourner des critères de soutenabilité pertinents (du point de vue de la mise en œuvre), compréhensibles et utilisables. Enfin, il me semble aussi capital d'effectuer un travail sur les données (existence,

qualité, élaboration, méthodologie, outils etc.), ce qui ne peut être fait qu'en collaboration étroite avec tous nos partenaires.

Bibliographie complète d'Emmanuel Prados (juillet 2011)

[1] Peter Belhumeur, N., Katsushi Ikeuchi, Emmanuel Prados, Stefano Soatto, and Peter Sturm. Editorial for the Special Issue on Photometric Analysis for Computer Vision. *International Journal of Computer Vision*, 86(2-3) :125–126, 2010.

[2] Fabio Camilli and Emmanuel Prados. Shape-from-Shading with discontinuous image brightness. *Applied Numerical Mathematics*, 56(9) :1225–1237, 2006.

[3] Fabio Camilli and Emmanuel Prados. Viscosity Solution. In Katsushi Ikeuchi, editor, *the Encyclopedia of Computer Vision*. Springer, 2011.

[4] Amael Delaunoy, Ketut Fundana, Emmanuel Prados, and Anders Heyden. Convex Multi-Region Segmentation on Manifolds. In IEEE, editor, *The 12th IEEE International Conference on Computer Vision*, pages 662 – 669, Kyoto, Japon, 2009. IEEE.

[5] Amael Delaunoy and Emmanuel Prados. Gradient Flows for Optimizing Triangular Mesh-based Surfaces : Applications to 3D Reconstruction Problems dealing with Visibility. *International Journal of Computer Vision*, 2011.

[6] Amael Delaunoy, Emmanuel Prados, and Peter Belhumeur, N. Towards Full 3D Helmholtz Stereovision Algorithms. In Springer, editor, *Asian Conference on Computer Vision*, Queenstown, Nouvelle-Zélande, November 2010.

[7] Amael Delaunoy, Emmanuel Prados, Ketut Fundana, and Anders Heyden. Segmentation convexe multi-région de données sur les surfaces. In *17ème Congrès de Reconnaissance des Formes et Intelligence Artificielle*, Caen, France, January 2010.

[8] Amael Delaunoy, Emmanuel Prados, Pau Gargallo, Jean-Philippe Pons, and Peter Sturm. Stéréo multi-vues : erreur de reprojection et maillages triangulaires. In *ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur*, Trégastel, France, France, 2009.

[9] Amael Delaunoy, Emmanuel Prados, Pau Gargallo I Piracés, Jean-Philippe Pons, and Peter Sturm. Minimizing the Multi-view Stereo Reprojection Error for Triangular Surface Meshes. *British Machine Vision Conference*, 2008.

[10] Pau Gargallo, Emmanuel Prados, and Peter Sturm. Minimizing the Reprojection Error in Surface Reconstruction from Images. In *11th IEEE International Conference on Computer Vision (ICCV '07)*, pages 1–8, Rio de Janeiro, Brésil, 2007. IEEE Computer Society.

[11] Pau Gargallo, Emmanuel Prados, and Peter Sturm. Minimiser l'erreur de reprojection en reconstruction de surfaces basée images. In *RFIA'08*, Amiens, France, 2008.

[12] Byung-Woo Hong, Emmanuel Prados, Stefano Soatto, and Luminita Vese, Aura. Shape Representation based on Integral Kernels : Application to Image Matching and Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 833–840, New York, États-Unis, 2006. IEEE.

[13] Byung-Woo Hong, Emmanuel Prados, Luminita Vese, and Stefano Soatto. Shape Representation based on Integral Kernels : Application to Image Matching and Segmentation. Rapport Technique UCLA CSD-TR 050044, UCLA, 2005.

[14] INRIA, editor. *Proceedings Workshop on Photometric Analysis for Computer Vision (PACV'2007)*. INRIA, October 2007.

[15] Zsolt Janko, Amael Delaunoy, and Emmanuel Prados. Colour Dynamic Photometric Stereo for Textured Surfaces. In Springer, editor, *Asian Conference on Computer Vision*, Queenstown, Nouvelle-Zélande, November 2010.

[16] Hailin Jin, Daniel Cremers, Dejun Wang, Emmanuel Prados, Anthony Yezzi, and Stefano Soatto. 3-D Reconstruction of Shaded Objects from Multiple Images Under Unknown Illumination. *International Journal of Computer Vision*, 76(3), 2008.

[17] Christophe Lenglet, Emmanuel Prados, Jean-Philippe Pons, Rachid Deriche, and Olivier Faugeras. Brain Connectivity Mapping using Riemannian Geometry, Control Theory and PDEs. *SIAM Journal on Imaging Sciences*, 2(2) :285–322, 2009.

[18] Emmanuel Prados. Une approche du "Shape From Shading" par solutions de viscosité. D.e.a. master thesis, Université de Nice - Sophia Antipolis, Sophia Antipolis, September 2001.

[19] Emmanuel Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem*. These, Université de Nice Sophia-Antipolis, October 2004. Luis Alvarez University of Las Palmas (Spain),Pierre Bernhard University of Nice-Sophia Antipolis (France),Rachid Deriche INRIA Sophia Antipolis (France),Jean denis Durou University of Toulouse (France),Luc robert Realviz S.A.,Christoph Schnörr University of Mannheim (Germany),Olivier Faugeras (invited) INRIA Sophia Antipolis (France).

[20] Emmanuel Prados and Elise Arnaud. Modélisation numérique : Quel développement durable ? *La Recherche. Les Cahiers de l'Inria*, 2010(445 octobre 2010), October 2010.

[21] Emmanuel Prados, Elise Arnaud, Pierre-Yves Longaretti, and Francois Mancebo. Mathematical and numerical analyses of local integrated models. In *Workshop on Decision Analysis and Sustainable Development*, Montreal, Canada, September 2010.

[22] Emmanuel Prados, Elise Arnaud, Pierre-Yves Longaretti, and Francois Mancebo. Presentation of the SOCLE3 project and the STEEP lab. In *72nd meeting of the European Working Group "Multiple Criteria Decision Aiding"*, Paris, France, October 2010.

[23] Emmanuel Prados, Elise Arnaud, Francois Mancebo, and Pierre-Yves Longaretti. Des outils informatiques d'aide à la décision en faveur du Développement Durable. *Actu-Environnement.com*, October 2010.

[24] Emmanuel Prados, Fabio Camilli, and Olivier Faugeras. A viscosity method for Shape-from-Shading without boundary data. Technical Report RR-5296, INRIA, August 2004.

[25] Emmanuel Prados, Fabio Camilli, and Olivier Faugeras. De nouvelles "solutions" au problème du Shape from Shading. In *CANUM 2004 - 36th French Congress of Numerical Analysis*, Obernai, France, 2004.

[26] Emmanuel Prados, Fabio Camilli, and Olivier Faugeras. A unifying and rigorous Shape From Shading method adapted to realistic data and applications. *Journal of Mathematical Imaging and Vision*, 25(3) :307–328, 2006.

[27] Emmanuel Prados, Fabio Camilli, and Olivier Faugeras. A viscosity solution method for Shape-From-Shading without image boundary data. *Mathematical Modelling and Numerical Analysis*, 40(2) :393–412, 2006.

[28] Emmanuel Prados and Olivier Faugeras. A mathematical and algorithmic study of the Lambertian SFS problem for orthographic and pinhole cameras. Technical Report RR-5005, INRIA, November 2003.

[29] Emmanuel Prados and Olivier Faugeras. Approximations numériques des solutions de viscosité de divers EDPs provenant du Shape from Shading. In *CANUM$\tilde{2}$003 - 35th French Congress of Numerical Analysis*, La Grande Motte, France, 2003.

[30] Emmanuel Prados and Olivier Faugeras. Perspective shape from shading and viscosity solutions. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, pages 826–831, Nice, France, 2003. IEEE Computer Society.

[31] Emmanuel Prados and Olivier Faugeras. A rigorous and realistic Shape From Shading method and some of its applications. Technical Report RR-5133, INRIA, March 2004.

[32] Emmanuel Prados and Olivier Faugeras. Reconstruction photogramétrique des formes 3D ; nouveaux résultats théoriques et nouveaux algorithmes pour des projections orthographique et en perspective. In *Actes du 14ème Congrès de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France, 2004.

[33] Emmanuel Prados and Olivier Faugeras. Unifying Approaches and Removing Unrealistic Assumptions in Shape from Shading : Mathematics Can Help. In Jiri Matas Tomás Pajdla, editor, *Proceedings of the 8th European Conference on Computer Vision*, volume 3024 of *Lecture Notes in Computer Science*, pages 141–154, Prague, Czech Republic, Europe, 2004. Springer.

[34] Emmanuel Prados and Olivier Faugeras. Viscosity solutions for realistic Shape-From-Shading. , 2004.

[35] Emmanuel Prados and Olivier Faugeras. A generic and provably convergent Shape-From-Shading Method for Orthographic and Pinhole Cameras. *International Journal of Computer Vision*, 65(1/2) :97–125, 2005.

[36] Emmanuel Prados and Olivier Faugeras. Shape from Shading : a well-posed problem ? In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'05*, volume 2, pages 870–877, San Diego, États-Unis, 2005. IEEE.

[37] Emmanuel Prados and Olivier Faugeras. Rôle clé de la Modélisation en Shape From Shading. In *15ème Congrès de Reconnaissance des Formes et Intelligence Artificielle*, Tours, France, 2006.

[38] Emmanuel Prados and Olivier Faugeras. Shape From Shading. In Y. Chen N. Paragios and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 375–388. Springer, 2006.

[39] Emmanuel Prados, Olivier Faugeras, and Fabio Camilli. A mathematical framework unifying various Shape from Shading approaches. In *Mathematics and Image Analysis (MIA '04)*, Paris, France, 2004.

[40] Emmanuel Prados, Olivier Faugeras, and Fabio Camilli. Shape from Shading : a well-posed problem ? Technical Report RR-5297, INRIA, August 2004.

[41] Emmanuel Prados, Olivier Faugeras, and Elisabeth Rouy. Shape from Shading and Viscosity Solutions. In *European Conference on Computer Vision, ECCV 2002*, volume 2351 of *Lecture Notes in Computer Science*, pages 790–804, Copenhagen, Danemark, 2002. Springer.

[42] Emmanuel Prados, Olivier Faugeras, and Elisabeth Rouy. Shape-from-Shading and Viscosity Solutions. Technical Report RR-4638, INRIA, November 2002.

[43] Emmanuel Prados, Nitin Jindal, and Stefano Soatto. A Non-Local Approach to Shape From Ambient Shading. In *2nd International Conference on Scale Space and Variational Methods in Computer Vision (SSVM'09)*, Lecture Notes in Computer Science series, pages 696–708, Voss, Norvège, 2009. Springer-Verlag.

[44] Emmanuel Prados, Christophe Lenglet, Jean-Philippe Pons, Nicolas Wotawa, Rachid Deriche, Olivier Faugeras, and Stefano Soatto. Control Theory and Fast Marching Methods for Brain Connectivity Mapping. Rapport de recherche RR-5845, INRIA, 2006.

[45] Emmanuel Prados, Christophe Lenglet, Jean-Philippe Pons, Nicolas Wotawa, Rachid Deriche, Olivier Faugeras, and Stefano Soatto. Control Theory and Fast Marching Techniques for Brain Connectivity Mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1076–1083, New York, États-Unis, 2006.

[46] Emmanuel Prados and Stefano Soatto. Fast Marching Method for Generic Shape From Shading. In *3rd International Workshop on Variational, Geometric and Level Set Methods in Computer Vision, VLSM'05*, volume 3752 of *Lecture Notes in Computer Science*, pages 320–331, Beijing, Chine, 2005. Springer.

[47] Emmanuel Prados and Stefano Soatto. Méthode de "Fast Marching" générique pour le "Shape From Shading". In *15ème Congrès de Reconnaissance des Formes et Intelligence Artificielle*, Tours, France, 2006.

[48] Peter Sturm, Amael Delaunoy, Pau Gargallo, Emmanuel Prados, and Kuk-Jin Yoon. 3D and Appearance Modeling from Images. In Eduardo Bayro-Corrochano and Jan-Olof Eklundh, editors, *Proceedings of the 14th Iberoamerican Congress on Pattern Recognition - CIARP 2009*, volume 5856 of *Lecture Notes in Computer Science*, pages 695–704, Guadalajara, Mexique, 2009. Springer.

[49] Dejun Wang, Emmanuel Prados, and Stefano Soatto. Toward robust and physically plausible shaded stereoscopic segmentation. Rapport Technique UCLA CSD-TR 050013, UCLA, 2005.

[50] Dejun Wang, Emmanuel Prados, and Stefano Soatto. Towards Robust and Physically Plausible Shaded Stereoscopic Segmentation. In *Computer Vision and Pattern Recognition Workshop (CVPRW)*, New York, États-Unis, 2006. IEEE Computer Society.

[51] Nicolas Wotawa, Christophe Lenglet, Emmanuel Prados, Rachid Deriche, and Olivier Faugeras. Anatomical connections in the human visual cortex : validation and new insights using a DTI Geodesic Connectivity Mapping method. Rapport de recherche RR-6176, INRIA, 2007.

[52] Kuk-Jin Yoon, Emmanuel Prados, and Peter Sturm. Generic Scene Recovery using Multiple Images. In *2nd International Conference on Scale Space and Variational Methods in Com-*

puter Vision (SSVM'09), Lecture Notes in Computer Science series, Voss, Norvège, 2009. Springer.

[53] Kuk-Jin Yoon, Emmanuel Prados, and Peter Sturm. Joint Estimation of Shape and Reflectance using Multiple Images with Known Illumination Conditions. *International Journal of Computer Vision (IJCV)*, 86(2-3) :192–210, 2010. Korean Government (MOEHRD) (KRF-2006-352-D00087) and Flamenco project (grant ANR-06-MDCA-007).

[54] Kuk-Jin Yoon, Emmanuel Prados, Peter Sturm, Amael Delaunoy, and Pau Gargallo. Shape and Reflectance Recovery using Multiple Images with Known Illumination Conditions. Rapport de recherche RR-6309, INRIA, 2007.

Références

[55] http ://graphics.stanford.edu/data/3dscanrep/.

[56] T. de la. Barra. *Integrated Land Use and Transport Modelling*. Cambridge University Press, 1989.

[57] A. Borning and P. Waddell. Urbansim : interaction and participation in integrated urban land use, transportation, and environmental modeling. In Fortes and Macintosh [58], pages 133–134.

[58] J. A. B. Fortes and A. Macintosh, editors. *Proceedings of the 7th Annual International Conference on Digital Government Research, DG.O 2006, San Diego, California, USA, May 21-24, 2006*, volume 151 of *ACM International Conference Proceeding Series*. Digital Government Research Center, 2006.

[59] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3) :367–392, 2004.

[60] A. Hertzmann and S. M. Seitz. Example-based photometric stereo : Shape reconstruction with general, varying brdfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27 :1254–1264, August 2005.

[61] H. Jin, D. Cremers, A. J. Yezzi, and S. Soatto. Shedding light on stereoscopic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 36–42, 2004.

[62] H. Jin, D. Wang, D. Cremers, E. Prados, A. Yezzi, and S. Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 2007.

[63] H. Jin, A. J. Yezzi, and S. Soatto. Mumford-shah on the move : Region-based segmentation on deforming manifolds with application to 3-d reconstruction of shape and appearance from multi-view images. *J. Math. Imaging Vis.*, 29 :219–234, November 2007.

[64] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnoerr. Convex multi-class image labeling by simplex-constrained total variation. Technical report, University of Heidelberg, oct 2008.

[65] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[66] D. H. Meadows, D. L. Meadows, J. Randers, and W. W. Behrens. *Limits to Growth*. Signet, 1972.

[67] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, 72(2) :179–193, 2007.

[68] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[69] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints : a maximum-flow formulation. In *IEEE International Conference on Computer Vision*, pages 349–356, 2005.

[70] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 2007.

[71] Z. Sun, B. Deal, and V.G. Pallathucheril. The land-use evolution and impact assessment model : A comprehensive urban planning support system. *Journal of the Urban and Regional Information Systems Association*, 21(1) :57–68, 2009.

[72] A. Treuille, A. Hertzmann, and S. M. Seitz. Example-based stereo with general brdfs. In *ECCV (2)*, pages 457–469, 2004.

[73] G. Vogiatzis, C. Hernández Esteban, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12) :2241–2246, 2007.

[74] A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1) :31–43, 2003.

[75] A. Zaharescu, E. Boyer, and R.P. Horaud. Transformesh : a topology-adaptive mesh-based approach to surface evolution. In *In Proceedings of the Eighth Asian Conference on Computer Vision*, volume II of *LNCS 4844*, pages 166–175, Tokyo, Japan, nov 2007.

[76] T. Zickler. Reciprocal image features for uncalibrated helmholtz stereopsis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1801–1808, 2006.

[77] T. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis : Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2-3) :215–227, 2002.

[78] T.E. Zickler, J. Ho, D.J. Kriegman, J. Ponce, and P.N. Belhumeur. Binocular helmholtz stereopsis. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, volume 2, pages 1411–1417, oct 2003.

Deuxième partie

Recueil d’articles

Ce recueil compile mes articles les plus représentatifs en rapport direct avec les travaux décrits dans la partie I de ce document.

Voici leurs références bibliographiques complètes :

Paper 1 [10] : P. Gargallo, E. Prados, P. Sturm. Minimizing the Reprojection Error in Surface Reconstruction from Images. In *Proceedings of the International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil*, pages 1–8, October 14-20, 2007.

Paper 2 [53] : K.-J. Yoon, E. Prados, P. Sturm. Joint Estimation of Shape and Reflectance using Multiple Images with Known Illumination Conditions. In *the International Journal of Computer Vision* 86(2-3), pages 192–210, 2010.

Paper 3 [5] : A. Delaunoy, E. Prados. Gradient Flows for Optimizing Triangular Mesh-based Surfaces : Applications to 3D Reconstruction Problems dealing with Visibility. In *the International Journal of Computer Vision*, To Appear in 2011, already online.

Paper 4 [4] : A. Delaunoy, K. Fundana, E. Prados, A. Heyden. Convex Multi-Region Segmentation on Manifolds. In *Proceedings of the International Conference on Computer Vision, ICCV 2009*, pages 662–669, 2009.

Paper 5 [6] : A. Delaunoy, E. Prados, P. Belhumeur. Towards Full 3D Helmholtz Stereovision Algorithms. In *Proceedings of the Asian Conference on Computer Vision, ACCV 2010*, pages 39–52, 2010.

Paper 6 [43] : E. Prados, N. Jindal, S. Soatto. A Non-Local Approach to Shape From Ambient Shading. In *Proceedings of the International Conference on Scale Space and Variational Methods in Computer Vision, SSVM 2009, Voss, Norvège*, pages 696–708 , 2009.

Paper 7 [20] : E. Prados, E. Arnaud. Modélisation numérique : Quel développement durable ? Dans *La Recherche*, 445 octobre 2010.

Minimizing the Reprojection Error in Surface Reconstruction from Images

Pau Gargallo Emmanuel Prados Peter Sturm
INRIA Rhône-Alpes, France
name.surname@inrialpes.fr

Abstract

This paper addresses the problem of image-based surface reconstruction. The main contribution is the computation of the exact derivative of the reprojection error functional. This allows its rigorous minimization via gradient descent surface evolution. The main difficulty has been to correctly take into account the visibility changes that occur when the surface moves. A geometric and analytical study of these changes is presented and used for the computation of derivative.

Our analysis shows the strong influence that the movement of the contour generators has on the reprojection error. As a consequence, during the proper minimization of the reprojection error, the contour generators of the surface are automatically moved to their correct location in the images. Therefore, current methods adding additional silhouettes or apparent contour constraints to ensure this alignment can now be understood and justified by a single criterion: the reprojection error.

1. Introduction

Reconstructing scene models from images is the problem of inverting the image formation process. Many scenes can be well represented by a surface and some additional quantities describing, for example, reflectance properties, lighting conditions or sensor parameters. Such a model, allows to precisely describe how the images were generated from the surface. However, recovering the surface from the images is an old, incompletely solved, computer vision challenge.

In such a context, a solution to the problem would be a surface Γ such that the images generated from the model are most similar to the observed images (i.e. the data). This naturally yields to formulating the problem as the minimization of an error measure between the observed and predicted values of pixels, carried out, importantly, over all pixels in all input images. This is not a trivial task, as will be shown in the following.

For many image formation models, the predicted value of a pixel $\mathbf{u}$ depends only on the position of the point $\pi_\Gamma^{-1}(\mathbf{u})$ that is viewed in that pixel and possibly on its normal $\mathbf{n}(\pi_\Gamma^{-1}(\mathbf{u}))$. This point is the first collision between the viewing ray of $\mathbf{u}$ and the surface Γ or, if non existing, a point in the background B. The error measure between a predicted and an observed image is then of the form

$$E(\Gamma) = \int_{\mathcal{I}} g\big(\, \pi_\Gamma^{-1}(\mathbf{u}),\, \mathbf{n}(\pi_\Gamma^{-1}(\mathbf{u}))\,\big)\, d\mathbf{u}\,, \qquad (1)$$

where $\mathcal{I}$ is the set of all pixels in the image, $d\mathbf{u}$ is the area measure on the sensor's image plane, and[1] $g : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$ gives the error measure for the pixel $\mathbf{u}$. We call (1) the *reprojection error functional* and the objective of this paper is to find a method for minimizing it.

This functional class (1) is wide enough to cover many image-based surface reconstruction problems. In section 6, we illustrate an example application to multi-view stereo, where g measures the difference between the observed color of a pixel and the one predicted by the reconstruction. Another example would be the reconstruction from noisy range images [24], where g would measure the difference between the captured depth at $\mathbf{u}$ and the depth of $\pi_\Gamma^{-1}(\mathbf{u})$.

In the last years, great advances on the minimization of surface functionals have been made. Several works have addressed the minimization of the *weighted area functionals*. These are functionals of the form

$$A(\Gamma) = \int_\Gamma g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, d\sigma\,, \qquad (2)$$

where g is integrated on the surface and $d\sigma$ is the surface's area measure. The derivative of this functional has been found, allowing therefore its minimization via gradient descent surface evolution [5, 7, 20]. It has also been shown how to find the global minimum of some of these functionals via graph cuts [2, 11] and continuous max-flow [1].

The difference between the functionals (1) and (2), emanates from the fact that the first is an integral over the image domain, i.e. where the data lives, while the latter is an integral over the surface.

[1] $\mathbb{S}^2$ represents the unit sphere, i.e. the space of normals.

To benefit from the existing knowledge about the weighted area functional, one may try to rewrite the functional (1) as an integral over the surface by counting only the visible points [15, 19, 25]. This gives,

$$E(\Gamma) = -\int_{\Gamma\cup B} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\, \nu_\Gamma(\mathbf{x})\, d\sigma\ , \qquad (3)$$

where ν_Γ is the visibility function (giving 1 for an $\mathbf{x}$ that is visible and 0 otherwise, cf. section 4) and where the fact that $d\mathbf{u} = -\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\, \nu_\Gamma(\mathbf{x})\, d\sigma$ has been used. In order to count all the pixels in the image, the integral extends over the surface but also over the background surface B. This is assumed to be a distant, fixed surface whose projection covers the whole image and whose shape is irrelevant.

We observe that the integrand obtained by the conversion depends on $\mathbf{x}$ and $\mathbf{n}(\mathbf{x})$ as in (2), but also especially on the whole surface Γ, because of the visibility term. Hence, the reprojection error functional is not a weighted area functional and the existing methods for minimizing the weighted area functionals can not be applied.

The main contribution of this paper is the computation of the derivative of the reprojection error functional (section 5), allowing therefore its minimization via gradient descent. To do so, we first study the changes of visibility while a surface moves (section 4). We will particularly observe that contour generators have a strong influence on these changes. When a contour generator moves, some hidden parts of the surface or the background appear behind it and some visible parts disappear. The backprojection $\pi_\Gamma^{-1}(\mathbf{u})$ of the pixels at the corresponding apparent contour moves suddenly from one part of the surface to another. This has a strong effect on the predicted value of these pixels and therefore on the reprojection error and its derivative.

As a consequence, the correct gradient descent evolution of the reprojection error automatically favors and ensures the alignment of the apparent contours of the reconstructed surface with discontinuities present in the images. This alignment thus provides a generalization of the visual hull that takes into account all the apparent contours and not only the silhouettes (i.e. *outer* apparent contours). The experiments of section 6 will demonstrate this alignment in a particular application of the functional to multi-view stereo.

2. Related Work

Most state of the art surface reconstruction algorithms [17] use, at some point, a weighted area functional. The cost of a surface point is defined by a photo-consistency measure using the images where this point is visible. Not being possible to include the visibility in the functional itself, it has to be determined before evolving the surface. This can be done once and for all [8, 14, 23] or iteratively, alternating the computation of the visibility with the optimization of the functional [5, 15, 21].

Any method not including the visibility in the functional suffers, to some extend, of the *minimal surface bias* [1, 26]. This is a bias towards small surfaces. Its most notable effect is that the null surface has cost 0 and is therefore the global minimum. A softer effect is the tendancy of small and thin parts of the surface to disappear.

Palliatives have been proposed. Ballooning forces [23] pump the surface to avoid shrinkage and tend to get balloon like results [26]. Surface evolution methods [5, 15] rely implicitly on the fact that, for sufficiently textured surfaces, a wide local minimum exists close to a good reconstruction. Thus, the evolution will stop before shrinking too much. Visual hull based approaches constrain the surface to fill the silhouettes of the object in the images [8, 6, 18]; the bias is thus reduced, but only in parts of the surface that are close to the visual hull.

Stereoscopic segmentation [25] use the concept of oriented visibility [11] to include the visibility in a weighted area functional. In consequence, the shrinkage is avoided and the resulting surface is consistent with the silhouettes in the images. This happens automatically without the need of additional constraints. However, the oriented visibility approximation is only valid for *convex* objects and the evolution derived in [25] does not correctly handle self-occlusions.

Visual hull constraints have been generalized to taking into account not only silhouettes but all apparent contours generators, by enforcing them to be aligned with strong image gradients [4, 9]. The same way that the stereoscopic segmentation manage to reconstruct visual hull like surfaces without silhouettes constraints, the proper minimization of the reprojection error presented in this paper performs the aligment of all the apparent contours naturally, without any additional constaints.

3. Mathematical Background and Notation

The mathematical framework used in this paper is the one defined by Solem and Overgaard [20] in which shapes are implicitly represented by level set functions [12, 13]. For the convenience of the reader, we remind the related notions and notations required for understanding our work.

3.1. Level Set and Characteristic Functions

Given a level set function $\phi : \mathbb{R}^3 \to \mathbb{R}$, the set of points $\Omega = \{\mathbf{x} : \phi(\mathbf{x}) \leq 0\}$ is a *solid shape* [10] and its boundary $\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$ is an *oriented surface*. We say that ϕ is an *implicit representation* of Ω and that Ω is the *inside* of Γ. The outward normal vector of the surface can be computed from the implicit representation as $\mathbf{n} = \nabla\phi/|\nabla\phi|$.

The *characteristic function* of the shape, χ_Ω, evaluates to 1 inside the shape and 0 outside. It can easily be expressed in terms of ϕ and the Heaviside step function, H, as $\chi_\Omega = 1 - H(\phi)$. The gradient $\nabla\chi_\Omega$ of the characteristic function

can only be defined in the distributional sense [16]. For all test vector fields $\mathbf{w} : \mathbb{R}^3 \to \mathbb{R}^3$,

$$\int_{\mathbb{R}^3} \nabla\chi_\Omega \cdot \mathbf{w}\, d\mathbf{x} \equiv -\int_\Omega \nabla \cdot \mathbf{w}\, d\mathbf{x} = -\int_\Gamma \mathbf{w} \cdot \mathbf{n}\, d\sigma \ , \quad (4)$$

where the last term results from Gauss' divergence theorem. In other words, the distribution $\nabla\chi_\Omega$ computes the flux of $\mathbf{w}$ that is entering the shape. An expression in terms of ϕ can be obtained by the chain rule:

$$\nabla\chi_\Omega = -\nabla\phi\, \delta(\phi) \ , \quad (5)$$

where δ is the Dirac delta distribution.

3.2. Functional Derivatives

Let M denote the *manifold of admissible surfaces* defined by Solem and Overgaard [20]. Points in this space are surfaces. The tangent vectors on a point Γ are the normal velocities by which the surface can evolve. The tangent space $T_\Gamma M$ is the set of all these normal velocities.

The variation $\phi^s = \phi + s\psi$ of ϕ, describes a curve $\Gamma(s) = \{\mathbf{x} : \phi^s(\mathbf{x}) = 0\}$ in M. Its *normal velocity* (or tangent vector) at $s = 0$ is,

$$v = \frac{-\psi}{|\nabla\phi|} \ . \quad (6)$$

Any tangent vector can be obtained in this way.

Consider a surface functional $E : M \to \mathbb{R}$. When the function $E(\Gamma(s))$ is derivable at $s = 0$ we say that the *Gâteaux derivative* of E at Γ in the direction v is

$$\partial E(\Gamma, v) \equiv \frac{d}{ds} E(\Gamma(s))\Big|_{s=0} \ . \quad (7)$$

If this derivative can be written as

$$\partial E(\Gamma, v) = \int_\Gamma w(\mathbf{x})\, v(\mathbf{x})\, d\sigma \ , \quad (8)$$

with w in the tangent space, we say that w is the gradient of E. This allows to evolve the surface in the direction, $-w$, which ensures a decrease of the functional. Indeed, if $\Gamma(t)$ satisfies $\frac{\partial}{\partial t}\Gamma = -w$, then we have $\frac{\partial}{\partial t}E(\Gamma(t)) = -\int_\Gamma w \cdot w\, d\sigma \leq 0$. Thus, an evolution in the direction $-w$ will decrease the functional.

4. Understanding the Visibility

This section presents an analysis of the visibility and its evolution. The analysis stands on the study of Tsai *et al.* [22], who described the dynamics of the visible regions as the observer moves. The goal here, is to compute the derivative of the visibility function with respect to surface variations instead.

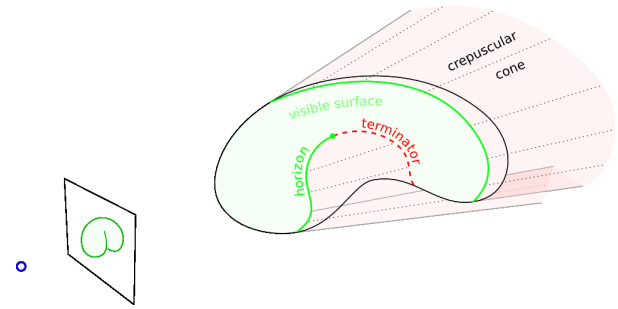


Figure 1. The *banana* shape seen from a vantage point. The horizon is drawn with green lines and the terminators with a dashed red line. The viewing ray segments are the crepuscular rays that form the crepuscular cone.

4.1. Geometrical Description

Given a solid shape and a vantage point, a point is said to be visible if no point of the shape lies on the segment between it and the vantage point. The set of all visible points will be called the *visible volume*, $\mathcal{V}$, and its complement, $\mathcal{V}^c$, the *occluded volume*. These volumes are respectively colored in green and red in Figure 2. The frontier between these two volumes, $\partial\mathcal{V}$, is a surface and will be called the *visibility interface*, see Figures 1 and 2.

The visibility interface is composed of two parts of different nature, one that is visible (the green curves in Figure 2) and one that is occluded (the red ones). The visible part of the interface coincide with the visible part of the shape's surface and background. Therefore, we will refer to this part as the *visible surface*. The occluded part of the interface is mostly in the free space. It is formed by patches of a generalized cone joining different parts of the visible surface. In analogy to atmospherical optics, we will call this part *crepuscular cone* (cf. also to Figure 1).

The border between the visible surface and the crepuscular cone is a closed curve on the shape's surface. Again, this curve contains both visible and occluded points. The visible part is the *horizon* or *contour generator* (green curves in Fig. 1). It is a (possibly open) curve made of visible surface points whose normal is perpendicular to the viewing ray. Its projection into the image are the *apparent contours*. The occluded part is the *terminator* (red curves in Fig. 1). It contains the points where the shadow of the horizon is cast. The segments joining points in horizon with their terminators are the *crepuscular rays* that form the crepuscular cone. The points in the crepuscular cone are all occluded by the horizon.

4.2. Mathematical Formulation

Let the visibility function $\nu_\Gamma : \mathbb{R}^3 \to \{0, 1\}$ be the characteristic function of the visible volume, the binary function evaluating to 1 for points that are visible and to 0 elsewhere.

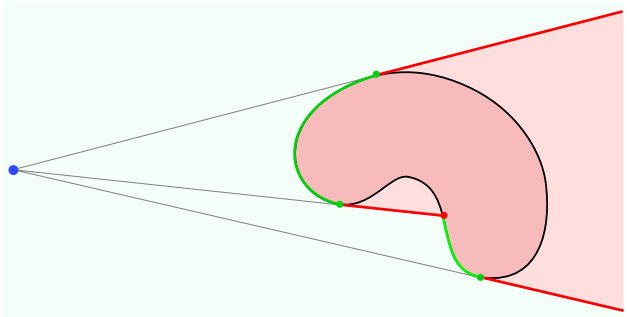


Figure 2. Sliced view of the banana shape. The visible surface/volume is drawn in green and the crepuscular cone/occluded volume in red.

In this section we write this function in terms of the level set function ϕ in order to derive analytical expressions for its spatial and temporal derivatives in the next sections. Later, the results will be used for computing the Gâteaux derivative of the reprojection error functional.

Assume the vantage point to be at the origin and let ϕ be an implicit representation of the surface. The visibility of a point $\mathbf{x}$ can be determined from the values that ϕ takes along the segment connecting the origin with $\mathbf{x}$. If any of these values is negative, then $\mathbf{x}$ is occluded.

Let $\mathbf{y}_\phi(\mathbf{x})$ be the point of the segment where ϕ admits the minimum (see Figure 3), i.e. $\mathbf{y}_\phi(\mathbf{x}) = \alpha_\phi(\mathbf{x})\mathbf{x}$ with

$$\alpha_\phi(\mathbf{x}) = \arg\min_{\alpha\in[0,1]} \phi(\alpha\mathbf{x}) \ . \tag{9}$$

If the minimum is not unique, take the closest to the origin. We observe that $\phi(\mathbf{y}_\phi(\mathbf{x}))$ is negative in the interior of the occluded volume and positive in the interior of the visible volume (cf. Figure 3). It can be shown that $\phi \circ \mathbf{y}_\phi$ is a continuous function [22], therefore, $\phi(\mathbf{y}_\phi(\mathbf{x})) = 0$ for all the points on the visibility interface regardless of their visibility.

This implies that for every point $\mathbf{x}$ on the visibility interface, $\mathbf{y}_\phi(\mathbf{x})$ is a point on the surface. If $\mathbf{x}$ is itself on the visible surface then necessarily $\mathbf{y}_\phi(\mathbf{x}) = \mathbf{x}$, otherwise $\mathbf{x}$ would be occluded. If $\mathbf{x}$ is on a crepuscular ray, then $\mathbf{y}_\phi(\mathbf{x})$ is its occluder, lying on the horizon where the crepuscular ray begins. All points on a crepuscular ray share the same occluder, $\mathbf{y}_\phi$. This fact gives an important role to the horizon and its consequences will be seen in the following sections.

From above, it follows that $\phi \circ \mathbf{y}_\phi$ is an implicit representation of the closure of the occluded volume and

$$\nu_\Gamma(\mathbf{x}) = H(\phi(\mathbf{y}_\phi(\mathbf{x}))) \tag{10}$$

almost everywhere, with exactly the exception of the visibility interface. Also, as distributions, $\nu_\Gamma = H \circ \phi \circ \mathbf{y}_\phi$.

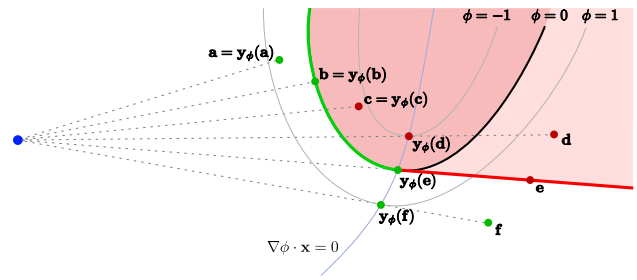


Figure 3. Location of $\mathbf{y}_\phi(\mathbf{x})$ according to the position of $\mathbf{x}$. If $\mathbf{x}$ is in the visible volume then $\phi(\mathbf{y}_\phi(\mathbf{x}))$ is positive, else it is negative.

4.3. Spatial Derivative of the Visibility

The gradient of the visibility, $\nabla\nu_\Gamma$, is a distribution that computes flow integral across the visibility interface $\int \nabla\nu_\Gamma \cdot \mathbf{w} d\mathbf{x} = \int_{\partial\mathcal{V}} \mathbf{w} \cdot d\boldsymbol{\sigma}$ (see section 3.1). It can be imagined as vector field that is zero everywhere except on the visibility interface, where it is aligned with the interface's normal and is infinitely long.

In order to derive an analytical expression of the gradient $\nabla\nu_\Gamma$, we first note that

$$\nabla(\phi \circ \mathbf{y}_\phi) = \nabla\phi(\mathbf{y}_\phi)\alpha_\phi \tag{11}$$

almost everywhere. Specifically, this holds for all the points on the visibility interface except for the terminators where $\phi \circ \mathbf{y}_\phi$ is not derivable. To see this, we distinguish two cases. If $\mathbf{y}_\phi$ is in the interior of the segment between vantage point and $\mathbf{x}$, then it is a local minimum of ϕ in the ray and thus $\nabla\phi(\mathbf{y}_\phi(\mathbf{x})) \cdot \mathbf{x} = 0$. As a consequence, the chain rule yields the above result. Otherwise, when $\mathbf{y}_\phi$ is at an extremum of the segment, we generally have that $\mathbf{y}_\phi(\mathbf{x}) = \mathbf{x}$ and $\alpha_\phi(\mathbf{x}) = 1$ in a neighborhood of $\mathbf{x}$ and so the same consequence holds.

Now, applying the chain rule to (10) it follows that

$$\nabla\nu_\Gamma = \delta(\phi(\mathbf{y}_\phi))\ \nabla\phi(\mathbf{y}_\phi)\ \alpha_\phi \ . \tag{12}$$

4.4. Temporal Derivative of the Visibility

Consider a variation $\phi^s = \phi + s\psi$ of ϕ and let $\Gamma(s)$ be the associated deformed surface. The visibility function $\nu_{\Gamma(s)}(\mathbf{x})$ is now a space-time function. Its derivative with respect to time is a scalar distribution concentrated on the visibility interface. It measures the variation of quantities integrated over the visible domain. Intuitively, that is the difference between the amount of mass that enters and that exits the visible volume as the surface evolves.

The chain rule gives

$$\frac{d}{ds}\nu_{\Gamma(s)}(\mathbf{x})\Big|_{s=0} = \delta(\phi(\mathbf{y}_\phi))(\psi(\mathbf{y}_\phi) + \nabla\phi(\mathbf{y}_\phi)\cdot\dot{\mathbf{y}}_\phi) \tag{13}$$

where $\dot{\mathbf{y}}_\phi$ is the temporal derivative of $\mathbf{y}_{\phi^s}$ at $s = 0$. If $\mathbf{y}_\phi$ is in the interior of the segment, then $\dot{\mathbf{y}}_\phi$ and $\mathbf{x}$ are collinear

and orthogonal to $\nabla\phi(\mathbf{y}_\phi)$. Otherwise, if $\mathbf{y}_\phi = \mathbf{x}$ then $\dot{\mathbf{y}}_\phi = 0$. So, in any case, we have

$$\frac{d}{ds}\nu_{\Gamma(s)}(\mathbf{x})\Big|_{s=0} = \delta(\phi(\mathbf{y}_\phi))\,\psi(\mathbf{y}_\phi)\,. \tag{14}$$

4.5. Temporal Derivative of a Quantity Integrated over the Visible Volume

We have now the necessary tools to compute the Gâteaux derivative of a functional F that is the integral of a quantity f over the visible volume,

$$F(\Gamma) = \int_{\mathbb{R}^3} f(\mathbf{x})\nu_\Gamma(\mathbf{x})d\mathbf{x}\,. \tag{15}$$

This derivative will be used in the following section to easily derive the derivative of the reprojection error functional. The main difficulties are contained in this section.

From equation (14), we see that

$$\begin{aligned}\frac{d}{ds}F(\Gamma(s))\Big|_{s=0} &= \int_{\mathbb{R}^3} f(\mathbf{x})\frac{d}{ds}\nu_{\Gamma(s)}(\mathbf{x})\Big|_{s=0}d\mathbf{x}\\ &= \int_{\mathbb{R}^3} f(\mathbf{x})\psi(\mathbf{y}_\phi)\delta(\phi(\mathbf{y}_\phi))\,d\mathbf{x}\,,\end{aligned} \tag{16}$$

which we rewrite as an integral over the visibility interface

$$\frac{d}{ds}F(\Gamma(s))\Big|_{s=0} = \int_{\partial\mathcal{V}} f(\mathbf{x})\frac{\psi(\mathbf{y}_\phi)}{|\nabla(\phi\circ\mathbf{y}_\phi)|}\,d\sigma\,, \tag{17}$$

by noting that $d\sigma = |\nabla(\phi\circ\mathbf{y}_\phi)|\delta(\phi(\mathbf{y}_\phi))d\mathbf{x}$. Ideally, we would like to write the derivative as an integral over the surface Γ and not over the visibility interface (see section 3.2). With this aim, we split the integral into a sum of two integrals, one over each of the parts of the visibility interface.

(i) On the visible surface, we know that $\mathbf{y}_\phi = \mathbf{x}$ and, thus, the integral is simply

$$\int_{\Gamma\cap\mathcal{V}} f(\mathbf{x})\frac{\psi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}d\sigma\,. \tag{18}$$

Note that, without loss of generality, we can assume ψ to be null on the background and, therefore, the integral over the background surface vanishes.

(ii) On the crepuscular cone, $\mathbf{y}_\phi$ is the occluder of $\mathbf{x}$ and is a point on the horizon of the surface. All the points on a crepuscular ray share the same occluder on the horizon. The idea, here, is to attribute all the mass of a crepuscular ray to the origin of the ray on the horizon. This way, the integral over the crepuscular cone will be written as an integral over the horizon and, therefore, over the surface Γ.

Given an arc parametrization of the horizon, $\boldsymbol{\gamma} : I \to \mathbb{R}^3 : t \mapsto \boldsymbol{\gamma}(t)$, the crepuscular cone can be parameterized by $\mathbf{x}(r,t) = r\boldsymbol{\gamma}(t)$ with $t \in I$ and r in the interval $(1, T_{\boldsymbol{\gamma}(t)})$; where for any fixed t, $\mathbf{x}(r,t)$ covers the crepuscular ray from the horizon $\boldsymbol{\gamma}(t)$ to the associated terminator. By using this parametrization, equation (11) and the fact that $\mathbf{y}_\phi(\mathbf{x}(r,t)) = \boldsymbol{\gamma}(t)$, the surface integral (17) over the crepuscular cone is written as

$$\int_I\int_1^{T_{\boldsymbol{\gamma}(t)}} f(r\boldsymbol{\gamma}(t))\frac{\psi(\boldsymbol{\gamma}(t))}{|\nabla\phi(\boldsymbol{\gamma}(t))|}r^2|\boldsymbol{\gamma}(t)\times\boldsymbol{\gamma}'(t)|\,dr\,dt\,. \tag{19}$$

The terms depending on r can be gathered together into $L(\mathbf{x}) = \int_1^{T_\mathbf{x}} f(r\mathbf{x})r^2dr$, which cumulates the mass of f along the crepuscular rays. Also, let $\boldsymbol{\eta}(t)$ denote the normal vector to the horizon, that is tangent to the surface and points away from the observer. The integral is, then,[2]

$$\int_I L(\boldsymbol{\gamma}(t))\frac{\psi(\boldsymbol{\gamma}(t))}{|\nabla\phi(\boldsymbol{\gamma}(t))|}\;(\boldsymbol{\gamma}(t)\cdot\boldsymbol{\eta}(t))\;dt\,. \tag{20}$$

This is the flux of the vector field $L\frac{\psi}{|\nabla\phi|}\mathbf{x}$ crossing the horizon from the visible surface to the crepuscular cone. It is an integral over the horizon. Let us now convert it into an integral over the surface Γ.

Let $\mathcal{O} = \{\mathbf{x}\cdot\mathbf{n}(\mathbf{x}) \le 0 : \mathbf{x} \in \Gamma\}$ be the set surface points whose normal is oriented towards the camera. The border of this set as a subset of the surface, $\partial\mathcal{O}$, is a curve on the surface. The horizon corresponds exactly to the visible part of this curve. From (4) on the Riemanian manifold Γ (instead of $\mathbb{R}^3$), we know that

$$\int_\Gamma \nabla_\Gamma\chi_\mathcal{O}\cdot\mathbf{w}\,d\sigma = -\int_{\partial\mathcal{O}}\mathbf{w}\cdot\boldsymbol{\eta}\,d\tau\,, \tag{21}$$

where ∇_Γ denotes the intrinsic gradient in Γ and $\chi_\mathcal{O} = 1-H(\mathbf{x}\cdot\mathbf{n})$ is the characteristic function of $\mathcal{O}$. Considering the vector field $\mathbf{w}(\mathbf{x}) = \nu_\Gamma(\mathbf{x})L(\mathbf{x})\frac{\psi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\mathbf{x}$, which is zero for all the points of $\partial\mathcal{O}$ except the horizon, we have that equation (20) can be written as

$$\int_\Gamma \nu_\Gamma(\mathbf{x})L(\mathbf{x})\frac{\psi(\mathbf{x})}{|\nabla\phi(\mathbf{x})|}\;(\mathbf{x}\cdot\nabla_\Gamma[H(\mathbf{x}\cdot\mathbf{n})])\,d\sigma\,. \tag{22}$$

Since $\nabla[H(\mathbf{x}\cdot\mathbf{n})]$ is on the tangent plane of Γ, it corresponds to $\nabla_\Gamma[H(\mathbf{x}\cdot\mathbf{n})]$. Thus, by (5), we can rewrite $\mathbf{x}\cdot\nabla_\Gamma[H(\mathbf{x}\cdot\mathbf{n})]$ as $\mathbf{x}^t\nabla\mathbf{n}\mathbf{x}\;\delta(\mathbf{x}\cdot\mathbf{n})$. Finally, joining the splited integrals (18) and (22), the Gâteaux derivative of F is

$$\partial F(\Gamma, v) = \int_\Gamma -\big(f + L\,\mathbf{x}^t\nabla\mathbf{n}\mathbf{x}\,\delta(\mathbf{x}\cdot\mathbf{n})\big)\,\nu_\Gamma\;v\;d\sigma. \tag{23}$$

Remark: As expected in section 3.2, we have managed to rewrite the Gâteaux derivative of F as $\int_\Gamma w\;v\;d\sigma$. Nevertheless, unusually here, w is not a function, but a distribution. Distributions are linear continuous operators, so the functional is Fréchet differentiable and the differential is w. However, the gradient in the tangent space, as defined in

[2] $\boldsymbol{\gamma}'$ and $\boldsymbol{\eta}$ form an orthonormal basis of the tangent plane at $\boldsymbol{\gamma}$. In this basis, $\boldsymbol{\gamma} = (\boldsymbol{\gamma}\cdot\boldsymbol{\gamma}')\boldsymbol{\gamma}' + (\boldsymbol{\gamma}\cdot\boldsymbol{\eta})\boldsymbol{\eta}$ and thus $|\boldsymbol{\gamma}\times\boldsymbol{\gamma}'| = \boldsymbol{\gamma}\cdot\boldsymbol{\eta}$.

[20] and [3], does not exist for this functional, because w is not an admissible deformation of Γ. In other words, to perform a gradient descent evolution w has to be approximated by an admissible deformation. In practice, this reduces simply to approximating the delta distribution with a function.

5. Differential of the Reprojection Error

Using the results of the previous section, in this section we are going to compute the differential of the reprojection error functional (3),

$$E(\Gamma) = -\int_{(\Gamma\cup B)\cap\mathcal{V}} g(\mathbf{x},\mathbf{n}(\mathbf{x}))\,\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\,d\sigma\,. \tag{24}$$

Since, for all the points $\mathbf{x}$ of the visible surface, the normal $\mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})$ to the visibility interface $\partial\mathcal{V}$ coincides with the normal $\mathbf{n}(\mathbf{x})$ to the surface Γ, and since $\mathbf{x}\cdot\mathbf{n}_{\partial\mathcal{V}}(\mathbf{x}) = 0$ on the crepuscular cone, it follows that the integral can be extended to the whole visibility interface

$$E(\Gamma) = -\int_{\partial\mathcal{V}} g(\mathbf{x},\mathbf{n}(\mathbf{x}))\,\frac{\mathbf{x}}{\mathbf{x}_z^3}\cdot\mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})\,d\sigma\,. \tag{25}$$

5.1. Case where g does not depend on the Normal

Let us first consider the case where g does not depend on the normal, $g(\mathbf{x},\mathbf{n}(\mathbf{x})) = g(\mathbf{x})$. The functional (25) is the flux of the vector field $g(\mathbf{x})\frac{\mathbf{x}}{\mathbf{x}_z^3}$ across the visibility interface $\partial\mathcal{V}$. By Gauss' divergence theorem, this flux is the opposite of the amount of divergence of the vector field inside the visible volume. Thus, as $\nabla\cdot\frac{\mathbf{x}}{\mathbf{x}_z^3} = 0$,

$$E(\Gamma) = \int_{\mathbb{R}^3}\left(\nabla g(\mathbf{x})\cdot\frac{\mathbf{x}}{\mathbf{x}_z^3}\right)\nu_\Gamma(\mathbf{x})\,d\mathbf{x}\,. \tag{26}$$

Now, by using the result (23) developed in the previous section with $f(\mathbf{x}) = \nabla g(\mathbf{x})\cdot\frac{\mathbf{x}}{\mathbf{x}_z^3}$, we immediately get the Gâteaux derivative of this functional. We observe that, in this case, L has a simple form, because the integral sums up the variations of g along the crepuscular rays. This is

$$L(\mathbf{x}) = \int_1^{T_\mathbf{x}} \nabla g(r\mathbf{x})\cdot\frac{\mathbf{x}}{\mathbf{x}_z^3}dr = \big[g(T(\mathbf{x}))-g(\mathbf{x})\big]\frac{1}{\mathbf{x}_z^3}\,, \tag{27}$$

where $T(\mathbf{x})$ is the terminator of $\mathbf{x}$. Finally, noting $g\circ T$ by g', the differential of the reprojection error functional is

$$-\nabla g\cdot\frac{\mathbf{x}}{\mathbf{x}_z^3}\nu_\Gamma + (g-g')\frac{\mathbf{x}^t\nabla\mathbf{n}\mathbf{x}}{\mathbf{x}_z^3}\delta(\mathbf{x}\cdot\mathbf{n})\nu_\Gamma. \tag{28}$$

5.2. With Normals

We describe here the derivation of the differential for the general case where g may depend on the normal of the surface. Because of space limitation, we will only sketch the calculus and present the final result.

First, using (4) with the visibility interface $\partial\mathcal{V}$ (instead of the surface Γ), we reformulate (25) as an integral over $\mathbb{R}^3$. Then, deriving the result with respect to s, the product rule yields

$$\begin{aligned}\frac{d}{ds}E(\Gamma(s))\Big|_{s=0} &= -\int_{\mathbb{R}^3}\frac{d}{ds}g(\mathbf{x},\mathbf{n}^s)\Big|_{s=0}\frac{\mathbf{x}}{\mathbf{x}_z^3}\cdot\nabla\nu_\Gamma\,d\mathbf{x}\\ &\quad -\int_{\mathbb{R}^3} g(\mathbf{x},\mathbf{n})\frac{\mathbf{x}}{\mathbf{x}_z^3}\cdot\nabla\frac{d}{ds}\nu_{\Gamma(s)}\Big|_{s=0}d\mathbf{x}\,.\end{aligned}$$

For the first integral, we have $\frac{d}{ds}g(\mathbf{x},\mathbf{n}^s)|_{s=0} = g_\mathbf{n}\cdot\frac{\nabla\psi}{|\nabla\phi|}$. To get rid of the $\nabla\psi$ term and make ψ appear instead, one has to do integration by parts on $\nabla\psi$, as done in [20], section 5. The second integral, also by integration by parts, becomes $-\int_{\mathbb{R}^3}\nabla\cdot[g(\mathbf{x},\mathbf{n})\frac{\mathbf{x}}{\mathbf{x}_z^3}]\frac{d}{ds}\nu_{\Gamma(s)}\big|_{s=0}d\mathbf{x}$ and so, as $\mathbf{n}$ does not depend on s, one can apply the result (28) of the simpler case when g does not depend on the normal.

The resulting differential is

$$-\nabla\cdot\Big(g_\mathbf{n}\frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} + g\frac{\mathbf{x}}{\mathbf{x}_z^3}\Big)\nu_\Gamma + (g-g')\frac{\mathbf{x}^t\nabla\mathbf{n}\,\mathbf{x}}{\mathbf{x}_z^3}\delta(\mathbf{x}\cdot\mathbf{n})\nu_\Gamma\,. \tag{29}$$

5.3. Comparison with the Weighted Area

If we denote $\overline{g}(\mathbf{x},\mathbf{n}) = g(\mathbf{x},\mathbf{n})\frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3}$, then $[\overline{g}_\mathbf{n} + \overline{g}\ \mathbf{n}]$ corresponds to $[g_\mathbf{n}\frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} + g\frac{\mathbf{x}}{\mathbf{x}_z^3}]$. So the differential (29) is equal to the gradient of the weighted area functional $\int_\Gamma\overline{g}(\mathbf{x},\mathbf{n}(\mathbf{x}))\,d\sigma$ given in [7, 20], plus a new term $(g-g')\frac{\mathbf{x}^t\nabla\mathbf{n}\,\mathbf{x}}{\mathbf{x}_z^3}\delta(\mathbf{x}\cdot\mathbf{n})\nu_\Gamma$ which is due to the changes of visibility caused by the movement of the horizon.

6. Application to Multi-view Stereo

In this section we present a sample application of the reprojection error functional to multi-view stereo. To keep the example simple, the scene is assumed to be Lambertian and the illumination static. Note though that more elaborate reflectance models still lead to a reprojection error functional.

The image formation model is the same as the one of stereoscopic segmentation [25]. To explain the images, one needs a surface Γ and also the radiance of points of that surface and of the background. Let $C:\mathbb{R}^3\to\mathbb{R}^3$ be the radiance function that associates colors to the points of the 3D space and the background. Ideally, the color $I(\mathbf{u})$ observed at the pixel $\mathbf{u}$ of image I should be equal to the color of its backprojection onto the surface $C(\pi_\Gamma^{-1}(\mathbf{u}))$. Thus, the SSD reprojection error of the surface into an image is

$$E(\Gamma,C) = \int_\mathcal{I}\big(I(\mathbf{u}) - C(\pi_\Gamma^{-1}(\mathbf{u}))\big)^2\,d\mathbf{u}\,. \tag{30}$$

The reprojection error for a set of images is the sum of the individual reprojection errors. An additional smoothing area energy $\int_\Gamma d\sigma$ is also added to represent our prior belief that surfaces are smooth.

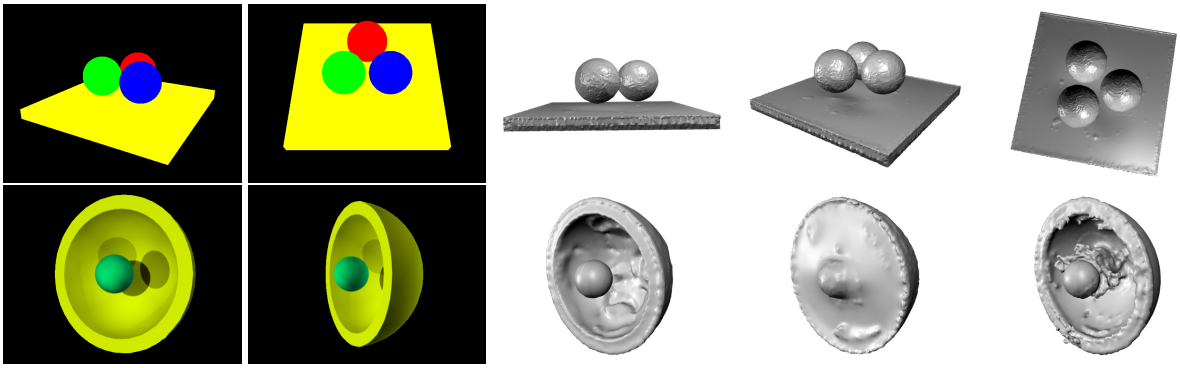


Figure 4. **Top:** two input images of the *balls* dataset and three rendereings of the reconstruction obtained with the horizon term. **Bottom:** two input images of the *bowl* dataset and the reconstruction obtained by the full term, the horizon term and the interior term.

The optimization is done by alternating between the estimation of C and Γ. For a fixed surface, the optimal radiance of a point has a closed form solution as a weighted sum of the colors observed at its projection onto the images where it is visible. For a fixed radiance, from equation (29), the differential of a single image error with respect to the surface is

$$(I-C)^t \nabla C \frac{\mathbf{x}}{\mathbf{x}_z^3} \nu_\Gamma + \left((I-C)^2 - (I-C')^2\right) \frac{\mathbf{x}^t \nabla \mathbf{n} \mathbf{x}}{\mathbf{x}_z^3} \delta(\mathbf{x}\cdot\mathbf{n}) \nu_\Gamma \tag{31}$$

where C' denotes the radiance at the terminator of $\mathbf{x}$.

Intuitively, this means that during the evolution, the visible points will move according to the first term of (31) in order to match C with I in the interior of objects in the image. Additionally, the second term will move the horizon of the surface and only the horizon because of the $\delta(\mathbf{x}\cdot\mathbf{n})\nu_\Gamma$ factor. This term compares the cost of the points in the horizon with the cost of the terminator and moves the horizon accordingly, so that the terminator becomes visible or occluded depending on that comparison. As a consequence, the apparent contours of the surface on the image will move to their correct location, as will be shown in the experiments.

6.1. Experiments

We implemented the surface evolution of (31) using the level set method in a multi-resolution scheme. Visibility is computed by rendering the surface using graphics hardware and then comparing the depth of the points with the Z-buffer. The horizons are found by approximating the delta distribution with a Gaussian. Radiance at terminators is computed by re-rendering the surface, horizons excluded.

We present here, the experiments performed on two, specially designed, synthetic scenes and two real world scenes. The goal of these experiments is to show the impact of the proper handling of the visibility, not to evaluate the performance of the generative model (30) presented above.

The *balls* dataset (fig. 4) consists of 20 images of three balls floating above a plane. There is no texture or shading in any part of the scene. Therefore, the only information present in the images are the apparent contours. In addition, because of self-occlusions between the balls and the plane, the silhouettes of the foreground are not sufficient to distinguish that the balls are three separate objects.

The reprojection error minimizing flow (31) was executed 3 times. First, using the flow as it is, then, using only its second term (the horizon term) and, finally, using only the first term (the interior term). The first two executions successfully managed to separate the three balls and obtained a correct reconstruction. The third one, did not separate the balls during the evolution and, due to the lack of texture, did shrink and disappear. The shrinkage did happen even when initializing from the ground truth.

We repeated the experiment for the *bowl* scene (fig. 4). The scene contains a green ball inside a yellow bowl with Lambertian shading. The execution with the full flow, correctly recovered the concavity of the bowl and the shape of the ball. The execution using only the horizon term did not carve the concavity at all. The execution with the interior term, did carve the concavity, but not completely, keeping the ball and the bowl linked together. This shows how the interior and the horizon terms worked together, the first one carving the concavity and the second one enforcing the apparent contour of the ball on the images.

Finally, we tested the evolution on the dino and temple datasets of the multi-view stereo database [17]. Numerical evaluation of the reconstructions performed by D. Scharstein and B. Curless can be found at `http://vision.middlebury.edu/mview/`. While the final reconstructions are not specially precise, probably due to the simplicity of the model (30), it is interesting to see the evolution itself (Figure 5). Two initial ellipsoids deform and *grow* according to the horizon term until fully explaining the input images. The evolution took 40 mins, the last 30 mins of which the surface reminded nearly steady.

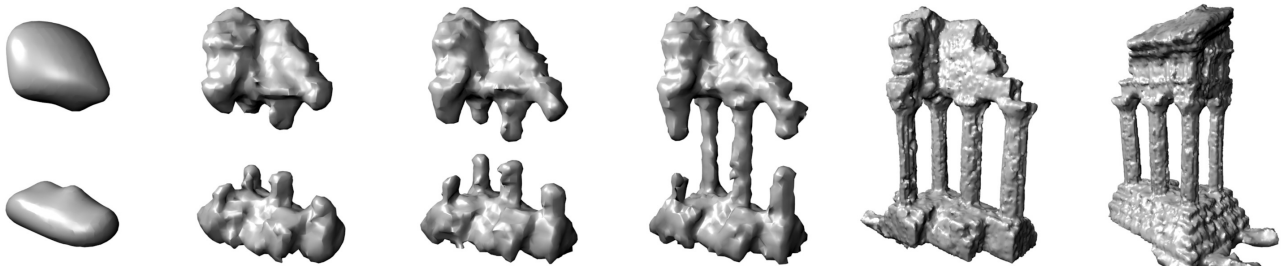


Figure 5. Surface evolution for the *temple sparse ring* dataset and the final reconstruction.

7. Conclusion

In this paper we compute the derivative of the reprojection error functional. The difficult part has been to correctly take into account the visibility changes that occur while the surface moves, which is one of the most challenging problems in surface reconstruction from images. The reward, is that it is now possible to minimize the reprojection error via surface evolution.

The benefit of this minimization is that the reconstructed surface is the one that best reproduces the observed images. In particular, as demonstrated in the experiments, the evolution moves the contour generators of the surface so that the apparent contours appear at their correct location in the images. This is a direct consequence of the correct minimization of the reprojection error itself. Therefore, current methods using additional silhouettes or apparent contour constraints can now be understood and justified by a single criterion: the reprojection error.

Acknowledgments The authors wish to thank Jean-Philippe Pons for sharing his level set method implementation and S.M. Seitz, B. Curless, J. Diebel, D. Scharstein and R. Szeliski for the temple and dino datasets and evaluations.

References

[1] B. Appleton, H. Talbot. Globally minimal surfaces by continuous maximal flows. *PAMI*, 28(1):106-118, 2006. 1, 2

[2] Y. Boykov, V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. *ICCV*, 2003. 1

[3] G. Charpiat, R. Keriven, J. P. Pons, O. Faugeras. Designing spatially coherent minimizing flows for variational problems based on active contours. *ICCV*, 2:1403–1408, 2005. 6

[4] D. Crispell, D. Lanman, P.G. Sibley, Y. Zhao, G. Taubin. Beyond Silhouettes: Surface Reconstruction using Multi-Flash Photography. *3DPVT*, 2006. 2

[5] O. Faugeras, R. Keriven. Complete dense stereovision using level set methods. *ECCV*, I:379–393, 1998. 1, 2

[6] Y. Furukawa, J. Ponce. Carved visual hulls for image-based modeling. *ECCV*, 2006. 2

[7] B. Goldlücke, I. Ihrke, C. Linz, M. Magnor. Weighted minimal hypersurface reconstruction. *PAMI*, 29(7):1194–1208, 2007. 1, 6

[8] C. Hernández Est., F. Schmitt. Silhouette and stereo fusion for 3D object modeling. *CVIU*, 96(3):367–392, 2004. 2

[9] R. Keriven. A variational framework for shape from contours. Tech. Report, CERMICS, ENPC, 2002-221b. 2

[10] J. Koenderink. *Solid shape*. MIT Press, 1990. 2

[11] V. Lempitsky, Y. Boykov, D. Ivanov. Oriented visibility for multiview reconstruction. *ECCV*, III:226–238, 2006. 1, 2

[12] S. Osher, J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. of Comput. Physics*, 79:12–49, 1988. 2

[13] S.J. Osher, R.P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002. 2

[14] S. Paris, F.X. Sillion, L. Quan. A surface reconstruction method using global graph cut optimization. *IJCV*, 66(2):141–161, 2006. 2

[15] J.P. Pons, R. Keriven, O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. *CVPR*, 2:822–827, 2005. 2

[16] W. Rudin. *Functional Analysis*. McGraw-Hill, 1991. 3

[17] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR*, 1:519–528, 2006. 2, 7

[18] S.N. Sinha, M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. *ICCV*, 1:349–356, 2005. 2

[19] S. Soatto, A.J. Yezzi, H. Jin. Tales of shape and radiance in multi-view stereo. *ICCV*, 974–981, 2003. 2

[20] J.E. Solem, N. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. *Scale-Space*, 419–430, 2005. 1, 2, 3, 6

[21] C. Strecha, R. Fransens, L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. *CVPR* 2004. 2

[22] Y.H.R. Tsai, L.T. Cheng, S. Osher, P. Burchard, G. Sapiro. Visibility and its dynamics in a PDE based implicit framework. *J. of Comput. Physics*, 199(1):260–290, 2004. 3, 4

[23] G. Vogiatzis, P. Torr, R. Cipolla. Multi-view stereo via volumetric graph-cuts. *CVPR*, 2:391–398, 2005. 2

[24] R.T. Whitaker. A Level-Set Approach to 3D Reconstruction from Range Data. *IJCV*, 29(3):203–231, 1998. 1

[25] A. Yezzi, S. Soatto. Stereoscopic segmentation. *IJCV*, 53(1):31–43, 2003. 2, 6

[26] T. Yu, N. Ahuja, W.-C. Chen. SDG cut: 3D reconstruction of non-lambertian objects using graph cuts on surface distance grid. *CVPR*, 2:2269–2276, 2006. 2

Int J Comput Vis (2010) 86: 192–210
DOI 10.1007/s11263-009-0222-4

Joint Estimation of Shape and Reflectance using Multiple Images with Known Illumination Conditions

Kuk-Jin Yoon · Emmanuel Prados · Peter Sturm

Received: 13 February 2008 / Accepted: 9 February 2009 / Published online: 6 March 2009
© Springer Science+Business Media, LLC 2009

Abstract We propose a generative model based method for recovering both the shape and the reflectance of the surface(s) of a scene from multiple images, assuming that illumination conditions and cameras calibration are known in advance. Based on a variational framework and via gradient descents, the algorithm minimizes simultaneously and consistently a global cost functional with respect to both shape and reflectance. The motivations for our approach are threefold. (1) Contrary to previous works which mainly consider specific individual scenarios, our method applies indiscriminately to a number of classical scenarios; in particular it works for classical stereovision, multiview photometric stereo and multiview shape from shading. It works with changing as well as static illumination. (2) Our approach naturally combines stereo, silhouette and shading cues in a single framework. (3) Moreover, unlike most previous methods dealing with only Lambertian surfaces, the proposed method considers general dichromatic surfaces. We verify the method using various synthetic and real data sets.

Keywords 3D reconstruction · Reflectance estimation · Multiview stereo · Photometric stereo · Multiview shape from shading

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-352-D00087) and by the FLAMENCO project (grant ANR-06-MDCA-007).

K.-J. Yoon (✉)
Room C508, Computer Vision Laboratory, Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), 261 Cheomdan-gwagiro (Oryong-dong), Buk-gu, Gwangju 500-712, Republic of Korea
e-mail: kjyoon@gist.ac.kr

E. Prados · P. Sturm
Perception Team, INRIA Grenoble, Montbonnot, Rhône Alpes, France

E. Prados
e-mail: Emmanuel.Prados@inrialpes.fr

P. Sturm
e-mail: Peter.Sturm@inrialpes.fr

1 Introduction and Related Work

Recovering the three-dimensional surface shape using multiple images is one of the major research topics in computer vision. Many methods have been proposed to solve the problem during these last two decades; refer to Seitz et al. (2006) for an evaluation of various recent methods. On the other hand, for a long time, the estimation of surface radiance/reflectance was secondary and was mainly of use to set up the shape reconstruction task (Faugeras and Keriven 1998; Zickler 2006; Zickler et al. 2002). Even some very recent works (Pons et al. 2005, 2007; Goesele et al. 2006; Zach et al. 2006; Tran and Davis 2006; Kolev et al. 2007b, 2007a) compute the 3D shape without considering radiance estimation. However, radiance/reflectance estimation has become a matter of concern in multiview reconstruction scenarios in the last decade. For example, Jin, Soatto et al. estimate conjointly the 3D shape and radiance (tensors) (see Jin et al. 2003, 2005; Soatto et al. 2003; Yezzi and Soatto 2003), or the 3D shape and the (piecewise constant) albedo of a Lambertian surface (Jin et al. 2008).

Here, radiance is a combination of lighting, surface reflectance, and the geometry of a scene. In other words, radiance contains shading and shadows and, from raw radiance, it is impossible to correct them when changing the lighting. Therefore, recovering reflectance is required for realistic relighting, which is also fundamental, for example, in virtual reality as well as augmented reality where the lighting con-

ditions when re-synthesizing a scene may be different from the lighting conditions when capturing a scene.

In addition, in real life applications, perfect Lambertian surfaces are rare and, therefore, multiview stereo algorithms have to be *robust to specular reflection*. Many ideas have been exploited to improve the robustness of the algorithms. A widespread idea is to use appropriate *similarity measures* as in Faugeras and Keriven (1998), Jin et al. (2002), Kim et al. (2003), Pons et al. (2005, 2007), Yang et al. (2003), Yoon and Kweon (2006). However, those similarity measures are not generally valid under general lighting conditions and/or not physically motivated. Another common strategy is to modify input images in order to *remove specular highlights* so as to obtain images as if the original surfaces had been purely Lambertian; see Yoon and Kweon (2006), Mallick et al. (2005), Zickler et al. (2008). These methods are based on the well known Neutral Interface Reflection (NIR) assumption (Lee et al. 1990) which supposes that the spectral energy distribution of specular reflection components is similar to the spectral energy distribution of incident light. Nevertheless, these methods are strongly limited by the specific lighting configuration. For example, the method in Yoon and Kweon (2006) is valid only for a single (uniformly) colored illumination. Although Zickler et al. (2008) recently showed that it is always possible to represent an image with $(M - N)$ specularity-independent color channels, where M is the number of color channels of an image and N is the number of different illuminant colors, their image representation may only work with up to two different illuminant colors because images have three color channels in general. Similarly, some authors do not consider the image pixels that potentially have specular reflection components—these are treated as outliers (Hernández Esteban et al. 2008; Birkbeck et al. 2006). The idea in this approach is to work only on data that one is able to model well (and so to ignore what is too complicated to model). The authors have then to increase the amount of data (i.e. the number of input images) in order to compensate for the loss of information. Thus, this strategy cannot be applied to two-frame stereo.

On the other hand, Bhat and Nayar (1998) analyzed the physics of specular reflection and the geometry of stereopsis to reduce errors due to non-Lambertian surfaces, which leads to a relationship between stereo vergence, surface roughness, and the likelihood of a correct match. Zickler et al. (2002) presented the Helmholtz stereopsis to overcome the specular reflection problem. However, these two approaches require *specialized camera/lighting configurations*. Concerning the robustness to non-Lambertian effects, it also worth to cite the work of Jin et al. (2005) which considers the so-called *radiance tensor*. However, although some similarity measures such as normalized cross correlation (Faugeras and Keriven 1998; Pons et al. 2005, 2007) could help to be robust to some illumination changes, the radiance tensor presented in Jin et al. (2005) is *not appropriate when images of the scene are taken under several different lighting conditions*.

In this paper, we propose a method for jointly estimating the shape and the reflectance of scene surfaces from multiple images. This can also be understood as the separation of geometry, reflectance, and illumination from radiance. In fact, our goal is to provide a shape and reflectance estimation method that is global (in the sense that it simultaneously and consistently optimizes shape and reflectance) and completely model based. *The method we propose is robust to non-Lambertian effects by directly incorporating a specular reflectance model* in the mathematical formulation of the problem. By incorporating a complete photometric image formation model, *it also advantageously exploits photometric phenomena*, as is explicitly done in photometric stereo methods. Furthermore, it allows to naturally deal with a set of images taken under several lighting conditions.

Some recent works already provide solutions in this direction. Goldman et al. (2005) present a relevant *photometric stereo* technique that simultaneously recovers shape and spatially-varying reflectance. They model spatially-varying reflectance as a linear combination of a small number of Ward BRDFs, while Hertzmann and Seitz (2005) use a similar representation of reflectance using images of simple objects made of the same material as the modeled scene. In addition, Yu et al. (2004, 2007) propose a model-based method for recovering the 3D shape and the reflectance of a non-Lambertian object. Nevertheless, in this last paper, the authors constrain the object to be made of a *single textureless material*; that is to say that the parameters of the reflectance (in particular the albedo) are the same for all the points of the object surface. So, the method in Yu et al. (2004, 2007) is a "multiview shape from shading" method, similarly as the one proposed by Jin et al. (2004, 2008) which focuses on the Lambertian case. To our knowledge, most works going in the same direction as ours are limited to surfaces made of a single (textureless) material. In particular, this is the case for the photometric stereo methods proposed by Georghiades (2003), Vogiatzis et al. (2005) and for the multiview photometric stereo work of Lu and Little (1995). Only a small number of similar works are able to recover scenes with varying albedo: Birkbeck et al. (2006) and Hernández Esteban et al. (2008). However, in these approaches, specular highlights are filtered out by using a simple thresholding. As a result, only diffuse components are used to estimate shape. Moreover, in Birkbeck et al. (2006), the authors simply compute the light visibility of a point using a surface normal and a light direction and Hernández Esteban et al. (2008) also used a thresholding to detect shadowed pixels that are not visible from light sources, which is however not working in the presence of multiple light sources. Finally, let us emphasize that the method of Hernández Esteban et al. (2008) is specifically a photometric stereo method, i.e. it

requires several different lighting configurations and cannot perform classical stereo-vision.

In our work, we do not want to restrain ourselves to a single textureless material, i.e. the reflectance properties of the object can be spatially variable. Actually, nowadays, more and more objects are printed and so it is fundamental to be able to recover textured and patterned objects. In return, of course, we will not be able to recover lighting conditions as done in Jin et al. (2004, 2008), and we have to use a separate process which computes them. In this work, we assume that lighting conditions are known in advance. In practice, we can use spherical objects with the reference white color to capture the direction and color of light sources (Powell et al. 2001; Zhou and Kambhamettu 2002).

More generally, one of the goals of this paper is to show that the joint computation of shape and reflectance is beneficial from several points of view. In addition to providing the reflectance of the scene (which is necessary e.g. for realistic re-lighting), this allows to naturally introduce specular models in the mathematical formulation of the multiview reconstruction problem; and thus this allows the method to be robust to highlights. Without any additional effort, this allows also to deal with a set of images lighted by several different conditions (which is not possible with radiance only). Moreover in such a case, the method allows to completely exploit the variations of the radiance according to the changes of illumination, as in photometric stereo. Finally, this enables to easily incorporate constraints on the reflectance and in particular to exploit shading effects in textureless regions (even if the number of different lighting conditions does not allow to do photometric stereo; e.g. if all images are taken under the same illumination).

Let us emphasize that, contrary to previous works that consider specific scenarios, our method can be applied indiscriminately to a number of classical scenarios—classical stereovision, multiview photometric stereo, and multiview shape from shading. Finally, based on the work of Gargallo et al. (2007), our method allows to naturally and simply combine in a single framework the three main cues available for shape reconstruction: silhouettes/apparent contours, stereo, and shading. To our knowledge, it is the first method which fuses these three cues in such a natural and convenient way. We do not claim that the method presented here gives better results (for 3D shape and reflectance) than previous approaches that are usually specific for certain scenarios: its intended merit is currently rather its generality.

The paper is organized as follows. In Sect. 2, we describe the modeling assumptions and we specify the notations used. In Sect. 3, we formulate the problem in the Bayesian framework; we then describe the associated cost functions in detail in Sect. 4. In Sect. 5, we precisely explain how we are minimizing the global energy. Experimental results on synthetic and real images data sets are shown in Sect. 6. Some discussion and future work are given in Sects. 7 and 8 concludes this paper.

2 Modeling Assumptions and Notations

We assume here that the scene can be decomposed into two entities: the foreground, which corresponds to the objects of interest, and the background. The foreground is composed by a set of (bounded and closed) 2D manifolds of $\mathbb{R}^3$. These surfaces are represented by S. More details are given below.

2.1 Cameras, Image Data, and Visibility

Image data are generated by n_c pinhole cameras. The perspective projection, from world to image coordinates, performed by the ith camera, is represented by $\Pi_i : \mathbb{R}^3 \to \mathbb{R}^2$. $\pi_i \subset \mathbb{R}^2$ is the image domain of the ith camera (i.e. the area covered by pixels). It is split into two parts: the pixels corresponding to the foreground, $\pi_{iF} = \pi_i \cap \Pi_i(S)$, and the other points $\pi_{iB} = \pi_i \setminus \pi_{iF}$ (associated to the background). $I_i : \pi_i \to \mathbb{R}^c$ is the image of the true scene, captured by the ith camera ($c = 1$ for a gray-scale image and $c = 3$ for a color image). We denote by I the set of input images: $I = \{I_1, I_2, \ldots, I_{n_c}\}$; I_{iF} and I_{iB} are the restrictions of the function I_i to π_{iF} and π_{iB}, respectively.

We consider the visibility function for image i, that is induced by the foreground surfaces S, $v_S^i : \mathbb{R}^3 \to \mathbb{R}$. It is defined as $v_S^i(\mathbf{X}) = 1$ if $\mathbf{X}$ is visible from the ith camera and $v_S^i(\mathbf{X}) = 0$ if it is occluded by the foreground. S_i denotes the part of S that is visible from the ith camera and $\Pi_{i,S}^{-1}$ is the back-projection from the ith camera onto S_i, i.e. for all points $\mathbf{x} \in \pi_{iF}$, $\Pi_{i,S}^{-1}(\mathbf{x})$ is the point on S along the ray joining $\mathbf{X}$ to the optical center of the ith camera, that is closest to the optical center.

2.2 Lighting Conditions

We model the illumination by a finite number of distant point light sources, together with an ambient illumination radiating constant energy isotropically in all directions. Illumination conditions may be different for every image. n_{il} is the number of illuminants corresponding to the ith image and $\mathbf{l}_{ij} \in \mathbb{S}^2$ and $L_{ij} \in \mathbb{R}^c$ are the direction and intensity[1] of the jth illuminant associated with the ith image, respectively. Similarly, $L_{ia} \in \mathbb{R}^c$ is the intensity[1] of the ambient illumination for the ith image.

To model occlusions of light sources from 3D points, we use light visibility functions $v_S^{i,j} : \mathbb{R}^3 \to \mathbb{R}$. We define $v_{L_{ij}}(\mathbf{X}) = 1$ if the jth illuminant of the ith image is visible from $\mathbf{X}$, $v_{L_{ij}}(\mathbf{X}) = 0$ otherwise. In addition, $S_{L_{ij}}$ is the part of S that is visible from the jth illuminant of the ith image.

[1] Non-normalized color vector, if $c = 3$.

In other words, $v_{L_{ij}}(\mathbf{X}) = 1$ if $\mathbf{X} \in S_{L_{ij}}$, $v_{L_{ij}}(\mathbf{X}) = 0$ otherwise. Based on the light visibility, we can take into account self-shadowing.

2.3 Modeling the Foreground Surface

We model the foreground object(s) by its shape S and its reflectance R. We denote $\Omega = (S, R)$. Contrary to most previous stereovision methods, we want to go beyond the Lambertian model. In order to get a solvable minimization problem without too many unknowns, we use a parametric reflectance model. Such a model should be selected in consideration of the applications aimed at. In this work, we consider the popular Blinn-Phong shading model (Blinn 1997). However, the proposed method is not limited to this model. By following the approach we describe, it is straightforward to use any other dichromatic reflection model.

We assume that $I_i(\mathbf{x})$ is equal to the radiance of the surface S at point $\mathbf{X} = \Pi_{i,S}^{-1}(\mathbf{x})$, in the direction of the ith camera. Thus, the images I_i can be decomposed as

$$I_i = I_{id} + I_{is} + I_{ia}, \tag{1}$$

where I_{id}, I_{is}, and I_{ia} are images containing the diffuse, specular, and ambient reflection components of I_i, respectively.

Diffuse reflectance is caused by the random scattering of light and it is independent of the viewing direction. By using the cosine law, this image component is described as

$$I_{id}(\mathbf{x}) = \sum_{j=1}^{n_{il}} v_{L_{ij}}(\mathbf{X})\big(\rho_d(\mathbf{X}) L_{ij}(\mathbf{n}(\mathbf{X}) \cdot \mathbf{l}_{ij})\big), \tag{2}$$

where $\rho_d(\mathbf{X}) \in \mathbb{R}^c$ is the diffuse albedo at point $\mathbf{X}$ and $\mathbf{n}(\mathbf{X})$ is the normal vector to the surface S at $\mathbf{X}$.

Specular reflectance is caused by the surface reflection, as with a mirror. This component is expressed as

$$I_{is}(\mathbf{x}) = \sum_{j=1}^{n_{il}} v_{L_{ij}}(\mathbf{X})\big(\rho_s(\mathbf{X}) L_{ij}(\mathbf{n}(\mathbf{X}) \cdot \mathbf{h}_{ij}(\mathbf{X}))^{\alpha_s(\mathbf{X})}\big), \tag{3}$$

where $\mathbf{h}_{ij}(\mathbf{X})$ is the bisector of the angle spanned by $\mathbf{X}$, the optical center of the ith camera and the jth illuminant. $\rho_s(\mathbf{X}) \in \mathbb{R}^c$ and $\alpha_s(\mathbf{X}) \in \mathbb{R}^+$ are the specular albedo and the shininess parameter at point $\mathbf{X}$.

The *ambient illumination* is assumed to be uniform in the scene and modeled as

$$I_{ia}(\mathbf{x}) = \rho_d(\mathbf{X}) L_{ia}, \tag{4}$$

where $\rho_d(\mathbf{X})$ is the diffuse albedo at $\mathbf{X}$, defined above, and L_{ia} is the intensity of the ambient illumination, defined in Sect. 2.2.

By combining the diffuse, specular, and ambient reflectance, we get the image formation equation as

$$I_i(\mathbf{x}) = \sum_{j=1}^{n_{il}} v_{L_{ij}}(\mathbf{X}) \mathbb{L}_{ij}(\mathbf{X}, \mathbf{n}(\mathbf{X})) + \rho_d(\mathbf{X}) L_{ia}, \tag{5}$$

where

$$\begin{aligned}\mathbb{L}_{ij}(\mathbf{X}, \mathbf{n}(\mathbf{X})) &= \mathbb{L}_{ij}^d(\mathbf{X}, \mathbf{n}(\mathbf{X})) + \mathbb{L}_{ij}^s(\mathbf{X}, \mathbf{n}(\mathbf{X})) \\ &= L_{ij}\rho_d(\mathbf{X})\big(\mathbf{n}(\mathbf{X}) \cdot \mathbf{l}_{ij}\big) \\ &\quad + L_{ij}\rho_s(\mathbf{X})\big(\mathbf{n}(\mathbf{X}) \cdot \mathbf{h}_{ij}(\mathbf{X})\big)^{\alpha_s(\mathbf{X})}.\end{aligned} \tag{6}$$

In the sequel, in order to simplify the notations, we denote $R = (R_d, R_s)$, where $R_d = \rho_d$ and $R_s = (\rho_s, \alpha_s)$.

2.4 Modeling the Background

For most works on multiview stereo, the cost functionals used attain their global optimum when the modeled surface shrinks to an empty set, inducing a *minimal surface bias*. This bias may be avoided as suggested by Yezzi and Soatto (2003) and Gargallo et al. (2007), by modeling the background of the scene, in addition to only the foreground objects. The choice of a background model is dictated by scenarios and applications. For example, in Yezzi and Soatto (2003), Jin et al. (2004), the background is characterized by its radiance which is constrained to be constant or strongly regular. On the other hand, when the background is quite irregular, one can assume to have background images, i.e. images captured by the same cameras, but without the foreground objects in the scene. In this work, we assume that, in addition to the input images I, we have these background images $\tilde{I} = \{\tilde{I}_1, \ldots, \tilde{I}_{n_c}\}$.[2] We also define $\tilde{I}_{iF}$ and $\tilde{I}_{iB}$, analogously to I_{iF} and I_{iB}.

3 Bayesian Formulation of the Problem

From a probabilistic point of view, the goal of this work is to estimate the shape S and the reflectance R of the foreground surface Ω, that maximize $P(\Omega|I)$ for given input images I. By Bayes' rule, the problem is then formulated as

$$\begin{aligned}P(\Omega|I) &= \frac{P(I|\Omega)P(\Omega)}{P(I)} \propto P(I|\Omega)P(\Omega) \\ &= P(I|S, R)P(S, R) \\ &= P(I|S, R)P(S)P(R)\end{aligned} \tag{7}$$

under the assumption that S and R are independent. Here,

[2] Another possibility is when the input are silhouette images, i.e. images where the foreground has already been segmented and the rest of the images been "painted" uniformly in some discriminative color. In this case, the background images are defined simply as completely uniform images of that color.

$P(I|\Omega) = P(I|S, R)$ is a likelihood and $P(S)$ and $P(R)$ are priors on shape and reflectance respectively.

3.1 Likelihood

When the camera calibration (i.e. the Π_i) and illumination conditions are given, we can produce synthetic images $\bar{I}_i(\Omega)$ corresponding to the input images I_i, by rendering the current estimate of Ω. The correct estimate of Ω will produce the same images, modulo noise and unmodeled effects of course. This allows us to measure the quality of the current estimate by comparing input images with rendered ones as in Schultz (1994), Yu et al. (2004). When assuming an independent identical distribution (i.i.d.) of measurement errors, the likelihood can be expressed as

$$\begin{aligned} P(I|\Omega) &\propto \prod_{i=1}^{n_c} \exp\big(-\xi_i(\Omega)\big) \\ &= \prod_{i=1}^{n_c} \exp\big(-\xi(I_i, \bar{I}_i(\Omega))\big), \end{aligned} \tag{8}$$

where $\xi_i(\Omega) = \xi(I_i, \bar{I}_i(\Omega))$ is a function of Ω, measuring the dissimilarity between two images I_i and $\bar{I}_i$.

3.2 Prior on Surface Shape S

A usual and plausible prior for the surface shape S concerns its area.[3] The prior can be expressed as

$$P(S) \propto \exp\big(-\psi(S)\big). \tag{9}$$

Here, $\psi(S)$ is the monotonic increasing function of the surface area $\int_S d\sigma$ where $d\sigma$ is the Euclidean surface measure.

3.3 Prior on Reflectance R

R is composed of two components, $R = (R_d, R_s)$. Here, unfortunately, reliably estimating specular reflectance for all surface points with only a uniform prior on its parameters, is very difficult unless there are observations of specular reflections for every surface point. For that reason, we need some specific prior on specular reflectance to be able to infer it in spite of the lack of such rich observations.[4]

It is physically valid to assume that specular reflectance varies smoothly within each homogeneous surface patch, i.e. that is made of the same material. This assumption is clearly reasonable in real life applications and in common scenes. It is, however, also very difficult to partition Ω according to the types of materials. In this work, we use the diffuse reflectance of the surface as a soft constraint to partition Ω and define the prior on the surface reflectance as

$$P(R) \propto \exp\big(-\omega(R)\big), \tag{10}$$

where $\omega(R)$ is a function of the intrinsic gradient of the diffuse and specular reflectance of a surface. This function is defined below, in Sect. 4.3.

4 Description of the Cost Functions

Based on the derivations in Sect. 3, the problem is formulated as

$$\begin{aligned} P(\Omega|I) &\propto P(I|\Omega)P(\Omega) = P(I|S, R)P(S, R) \\ &= P(I|S, R)P(S)P(R) \\ &\propto \prod_{i=1}^{n_c} \exp\big(-\xi_i(\Omega)\big) \times \big(\exp\big(-\psi(S)\big)\big) \\ &\quad \times \big(\exp\big(-\omega(R)\big)\big), \end{aligned} \tag{11}$$

and it can be expressed in terms of cost functions as

$$\begin{aligned} E_{total}(\Omega) &= E_{data}(\Omega) + E_{shape}(S) + E_{refl}(R) \\ &= \sum_{i=1}^{n_c} \xi_i(\Omega) + \psi(S) + \omega(R). \end{aligned} \tag{12}$$

Maximizing the probability (11) is equivalent to minimizing the total cost (12).

4.1 Data Cost Function

The current estimate of Ω gives a segmentation of each input image I_i into foreground I_{iF} and background I_{iB} and we can synthesize $\bar{I}_{iF}$ according to the image formation model of Sect. 2. As for $\bar{I}_{iB}$, it is generated according to the available background model. In this paper, as mentioned in Sect. 2.4, we use actual background images, i.e. $\bar{I}_{iB} = \tilde{I}_{iB}$. Also, as suggested by Yezzi and Soatto (2003), the similarity measure between observed and rendered images, $\xi_i(\Omega) = \xi(I_i, \bar{I}_i)$, is then rewritten as

$$\begin{aligned} \xi(I_i, \bar{I}_i) &= \xi_F(I_{iF}, \bar{I}_{iF}) + \xi_B(I_{iB}, \bar{I}_{iB}) \\ &= \xi_F(I_{iF}, \bar{I}_{iF}) + \xi_B(I_{iB}, \tilde{I}_{iB}) \\ &= \xi_F(I_{iF}, \bar{I}_{iF}) - \xi_F(I_{iF}, \tilde{I}_{iF}) \\ &\quad + \xi_F(I_{iF}, \tilde{I}_{iF}) + \xi(I_{iB}, \tilde{I}_i) \\ &= \hat{\xi}_F(I_{iF}, \bar{I}_{iF}) + \xi(I_i, \tilde{I}_i), \end{aligned} \tag{13}$$

[3] In this case, a minimal surface that may be characterized as the surface of minimal area under given boundary conditions will be sought.

[4] We will discuss some special cases that do not need any specific prior on the surface reflectance in Sect. 5.3.3.

where $\hat{\xi}_F(I_{iF}, \bar{I}_{iF}) = \xi_F(I_{iF}, \bar{I}_{iF}) - \xi_F(I_{iF}, \tilde{I}_{iF})$.[5] Since $\xi(I_i, \tilde{I}_i)$ is independent of Ω, the data cost function is written as

$$E_{data}(\Omega) = \sum_{i=1}^{n_c} \hat{\xi}_F(I_{iF}, \bar{I}_{iF}) + C, \tag{14}$$

where $C = \sum_{i=1}^{n_c} \xi(I_i, \tilde{I}_i)$ is constant.

4.1.1 Similarity Measure

When computing ξ, any statistical correlation between color or intensity patterns such as the sum of squared differences (SSD), cross correlation (CC), or mutual information (MI) can be used. In any case, ξ can be expressed as the integral over the image area as

$$\xi(I_i, \bar{I}_i) = \int_{\pi_i} e_i(\mathbf{x}) d\sigma_i, \tag{15}$$

where $d\sigma_i$ is the surface measure and $e_i(\mathbf{x})$ is the contribution at $\mathbf{x}$ to ξ_i. The data cost function is then given as

$$E_{data}(\Omega) = \sum_{i=1}^{n_c} \int_{\pi_{iF}} \hat{e}_i(\mathbf{x}) d\sigma_i + C, \tag{16}$$

where $\hat{e}_i(\mathbf{x}) = e_i(I_i(\mathbf{x}), \bar{I}_i(\mathbf{x})) - e_i(I_i(\mathbf{x}), \tilde{I}_i(\mathbf{x}))$. We adopt the derivations proposed in Pons et al. (2005) for ξ_i, e_i, and $\partial_2 e_i$.

4.1.2 Decoupling Appearance from Surface Normal

As shown in (5), surface appearance (i.e., the data cost function) is dependent on both the surface normal and the position, and this makes the problem hard to solve and unstable. To resolve this problem, we introduce an auxiliary photometric unit vector field $\mathbf{v}$ satisfying $\|\mathbf{v}\| = 1$ as in Jin et al. (2004), which is used for the computation of surface appearance. The vector field $\mathbf{v}$ is estimated in alternation with all other parameters, i.e. shape and reflectance. Equation (6) is written in terms of $\mathbf{v}$ as

$$\begin{aligned}\mathbb{L}_{ij}(\mathbf{X}, \mathbf{v}(\mathbf{X})) &= L_{ij}\rho_d(\mathbf{X})\big(\mathbf{v}(\mathbf{X}) \cdot \mathbf{l}_{ij}(\mathbf{X})\big) \\ &\quad + L_{ij}\rho_s(\mathbf{X})\big(\mathbf{v}(\mathbf{X}) \cdot \mathbf{h}_{ij}(\mathbf{X})\big)^{\alpha_s(\mathbf{X})},\end{aligned} \tag{17}$$

which is independent of $\mathbf{n}(\mathbf{X})$. To penalize the deviation between the actual normal vector $\mathbf{n}$ and the photometric normal vector $\mathbf{v}$, we add a new term

$$\begin{aligned}E_{dev}(\Omega) &= \tau \int_S \chi(\mathbf{X}) d\sigma = \frac{\tau}{2} \int_S \|\mathbf{n}(\mathbf{X}) - \mathbf{v}(\mathbf{X})\|^2 d\sigma \\ &= \tau \int_S (1 - (\mathbf{n}(\mathbf{X}) \cdot \mathbf{v}(\mathbf{X}))) d\sigma,\end{aligned} \tag{18}$$

to the cost function, where τ is a control constant.

4.2 Shape Area Cost Function

By using the area of a surface for the prior, the shape area cost function is simply defined as

$$E_{shape}(S) = \psi(S) = \lambda \int_S d\sigma, \tag{19}$$

where λ is a control constant.

4.3 Reflectance Discontinuity Cost Function

Based on the assumption on surface reflectance in Sect. 3.3, we define a discontinuity cost function of surface reflectance, which makes the discontinuities of specular reflectance generally coincide with the discontinuities of diffuse reflectance, as

$$E_{refl}(R) = \omega(R) = \beta \int_S f(\mathbf{X}) d\sigma, \tag{20}$$

where β is a control constant. $f(\mathbf{X})$ is defined as

$$f(\mathbf{X}) = \zeta\big(R_d(\mathbf{X})\big) \times \eta\big(R_s(\mathbf{X})\big), \tag{21}$$

where $\zeta(R_d(\mathbf{X}))$ and $\eta(R_s(\mathbf{X}))$ are defined in terms of the magnitude of the intrinsic gradients of diffuse reflectance and specular reflectance respectively as

$$\zeta\big(R_d(\mathbf{X})\big) = \left(1 - \frac{\|\nabla_S R_d(\mathbf{X})\|^2}{M}\right), \tag{22}$$

$$\eta\big(R_s(\mathbf{X})\big) = \big(\|\nabla_S \rho_s(\mathbf{X})\|^2 + \gamma \|\nabla_S \alpha_s(\mathbf{X})\|^2\big) \tag{23}$$

where M is a pre-defined constant and ∇_S denotes the intrinsic gradient defined on S. Here, $\zeta(R_d(\mathbf{X}))$ is inversely proportional to the magnitude of the intrinsic gradient of R_d.[6] In addition, $\eta(R_s(\mathbf{X}))$ is proportional to the magnitude of the intrinsic gradient of ρ_s and α_s.

By using the proposed discontinuity cost function of surface reflectance, surface points that do not have enough specular observations get assigned specular reflectance inferred from the specular reflectance of neighboring surface points with similar diffuse reflectance.

[5] Note that (13) is valid only when $\xi(I_i, \bar{I}_i)$ can be expressed as in (15).

[6] One has to use $M \geq 3$ for gray-level images and $M \geq 9$ for color images to make ζ positive.

4.4 Total Cost Function

By combining the cost functions defined in the previous sections, the total cost function is given by

$$\begin{aligned} E_{total}(\Omega) &= E_{data}(\Omega) + E_{dev}(\Omega) + E_{shape}(S) + E_{refl}(R) \\ &= C + \sum_{i=1}^{n_c} \int_{\pi_{iF}} \hat{e}_i(\mathbf{x}) d\sigma_i + \tau \int_S \chi(\mathbf{X}) d\sigma \\ &\quad + \lambda \int_S d\sigma \ + \ \beta \int_S f(\mathbf{X}) d\sigma. \end{aligned} \tag{24}$$

Here, it is worthy of notice that $E_{dev}(\Omega)$, $E_{shape}(S)$, and $E_{refl}(R)$ are defined over the scene surface while $E_{data}(\Omega)$ is defined as an integral over the image plane. By the change of variable

$$d\sigma_i = -\frac{\mathbf{d}_i(\mathbf{X}) \cdot \mathbf{n}(\mathbf{X})}{z_i(\mathbf{X})^3} d\sigma, \tag{25}$$

where $\mathbf{d}_i(\mathbf{X})$ is the vector connecting the center of the ith camera and $\mathbf{X}$ and $z_i(\mathbf{X})$ is the depth of $\mathbf{X}$ relative to the ith camera, we can replace the integral over the image plane by an integral over the surface as in Pons et al. (2007):

$$\begin{aligned} E_{data}(\Omega) &= C - \sum_{i=1}^{n_c} \int_{S_i} \left(\hat{e}_i(\Pi_i(\mathbf{X})) \frac{\mathbf{d}_i(\mathbf{X}) \cdot \mathbf{n}(\mathbf{X})}{z_i(\mathbf{X})^3} \right) d\sigma \\ &= C - \int_S \left(\sum_{i=1}^{n_c} v_S^i(\mathbf{X}) \hat{e}_i(\Pi_i(\mathbf{X})) \right. \\ &\quad \left. \times \frac{\mathbf{d}_i(\mathbf{X}) \cdot \mathbf{n}(\mathbf{X})}{z_i(\mathbf{X})^3} \right) d\sigma. \end{aligned} \tag{26}$$

As a result, the total cost function (24) is expressed as

$$\begin{aligned} E_{total}(\Omega) = C + \int_S \left(-\sum_{i=1}^{n_c} \left(v_S^i \hat{e}_i \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) \right. \\ \left. + \tau\chi + \lambda + \beta f \right) d\sigma. \end{aligned} \tag{27}$$

When denoting $g(\mathbf{X}, \mathbf{n}(\mathbf{X})) : \mathbb{R}^3 \times \Omega \to \mathbb{R}$ as

$$\begin{aligned} g(\mathbf{X}, \mathbf{n}(\mathbf{X})) = \left(-\sum_{i=1}^{n_c} \left(v_S^i \hat{e}_i \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) \right. \\ \left. + \tau\chi + \lambda + \beta f \right), \end{aligned} \tag{28}$$

Equation (24) is simply rewritten as

$$E_{total}(\Omega) = C + \int_S g(\mathbf{X}, \mathbf{n}(\mathbf{X})) d\sigma. \tag{29}$$

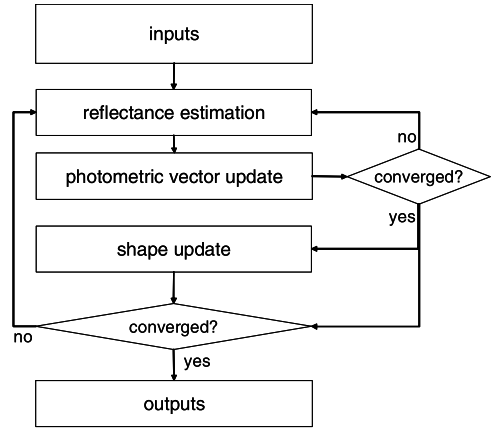


Fig. 1 Overall procedure of the proposed method. It is composed of three parts: shape update, reflectance estimation, and update of the auxiliary vector field **v**

Here, although the total cost function is an integral over the surface, it does not suffer from the usual minimal surface bias mentioned in Sect. 2.4: most functionals used in multiview stereo have an empty set as globally optimal surface, since they do not "explain" all pixels in the input images. Our approach, like Yezzi and Soatto (2003), takes into account all pixels in the cost function, not only those covered by the current estimate of the foreground object's shape, using both the estimated foreground and the available background information.

5 Scene Recovery

Recently, based on graph cuts or convexity, several global optimization methods have been proposed for the classical multiview stereovision problem, see Snow et al. (2000), Paris et al. (2006), Vogiatzis et al. (2007), Kolev et al. (2007b, 2007a). Nevertheless, because of the presence of the normal but also of the visibility in the cost function, the state of the art in optimization does not allow to compute the global minimum of the energy we have designed in the previous section. In this work, scene recovery is achieved by minimizing E_{total} via gradient descents.

In other respects, S and R are highly coupled and it is very complicated to estimate all unknowns simultaneously. To efficiently solve the problem, we adopt an alternating scheme, updating S for a fixed R and then R for a fixed S. This procedure is repeated until E_{total} no longer decreases and S and R no longer change. The overall procedure is shown in Fig. 1.

5.1 Shape Estimation—Surface Evolution

When assuming that R is given, E_{total} is a function of S. In this work, we derive the gradient descent flows correspond-

Int J Comput Vis (2010) 86: 192–210 199

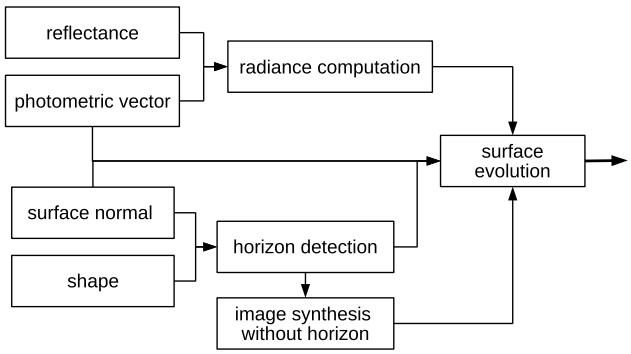


Fig. 2 Shape update

ing to the cost functions respectively. The final gradient descent flow is then given by

$$S_t = \left(S_t|_{data} + S_t|_{dev} + S_t|_{shape} + S_t g|_{refl}\right); \tag{30}$$

where $S_t|_{data}$, $S_t|_{dev}$, $S_t|_{shape}$ and $S_t\big|_{refl}$ are described below. The shape update scheme is shown in Fig. 2. For more details about gradient descent flows, we refer the inexperienced reader to Solem and Overgaard (2005) who nicely detail the geometric formulation of gradient descent in such a context.

5.1.1 Gradient Descent Flow for the Data Cost

As shown in (26), the data cost is a function of the visibility of a surface point, which is dependent on the whole surface shape, not only on the normal at the considered point. To correctly handle visibility for non-convex objects, self-occlusions of the foreground object must be taken into account, i.e. parts of the surface that occlude other parts of it from cameras. This is done according to Yezzi and Soatto (2003) and Gargallo et al. (2007) (which extends the work of Solem and Overgaard 2005), by writing $S_t|_{data}$ as

$$S_t|_{data} = \sum_{i=1}^{n_c}\left(-\frac{v_S^i(\hat{e}_i - \hat{e}_i')}{z_i^3}\left(\mathbf{d}_i^t \nabla \mathbf{n} \mathbf{d}_i^t \delta(\mathbf{d}_i \cdot \mathbf{n})\right) + \frac{v_S^i}{z_i^3}\left(\left(\partial_2 \hat{e}_i \nabla \bar{I}_i\right) \cdot \mathbf{d}_i\right)\right). \tag{31}$$

Here $\mathbf{d}_i$ is, as defined in Sect. 4.4, the vector connecting the center of the ith camera and a surface point $\mathbf{X}$, and $\delta(\cdot)$ is the delta function. Hence, the term $\delta(\mathbf{d}_i \cdot \mathbf{n})$ in the above expression is non-zero exactly at horizon points, i.e. points on (self-)occluding contours of the foreground. Let $\mathbf{X}'$ be the terminator of a horizon point $\mathbf{X}$ as shown in Fig. 3 (for more details see Gargallo et al. 2007; Gargallo 2008). $\hat{e}_i'$ is a similarity measure computed using the radiance of $\mathbf{X}'$ in the direction of the ith camera and the intensity at the image position corresponding to $\mathbf{X}$ and $\mathbf{X}'$ (cf. Sect. 4.1.1). Hence, the

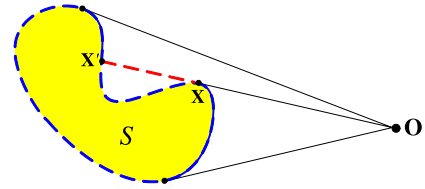


Fig. 3 Horizon point $\mathbf{X}$ and its terminator point $\mathbf{X}'$. $\mathbf{X}'$ is used to consider visibility changes. See Gargallo et al. (2007) for details

first term of the sum in (31) accounts for what happens to the likelihood when the visibility of non-convex objects changes due to the surface evolution. When a horizon point has no terminator point on the foreground surface itself, $\hat{e}_i' = 0$ because the terminator point is from the background. $\nabla \bar{I}_i$ is expressed by using (5) as

$$\nabla \bar{I}_i = \sum_{j=1}^{n_{il}} \{(\nabla v_{\mathbb{L}_{ij}})\mathbb{L}_{ij} + v_{L_{ij}}(\nabla \mathbb{L}_{ij})\} + (\nabla \rho_a) L_{ia}, \tag{32}$$

where

$$\nabla \mathbb{L}_{ij} = \nabla \mathbb{L}_{ij}^d + \nabla \mathbb{L}_{ij}^s, \tag{33}$$

and

$$\nabla \mathbb{L}_{ij}^d = L_{ij}(\nabla \rho_d)(\mathbf{v} \cdot \mathbf{l}_{ij}) + L_{ij}\rho_d\left(\nabla(\mathbf{v} \cdot \mathbf{l}_{ij})\right), \tag{34}$$

$$\nabla \mathbb{L}_{ij}^s = L_{ij}(\nabla \rho_s)(\mathbf{v} \cdot \mathbf{h}_{ij})^{\alpha_s} + L_{ij}\rho_s\left(\nabla(\mathbf{v} \cdot \mathbf{h}_{ij})^{\alpha_s}\right). \tag{35}$$

This gradient descent flow includes both the variation related to the camera visibility changes (the first term in (31)) and the variation related to the image changes (the second term in (31)), which also includes the variation due to the light visibility changes. Here, it is worthy of notice that the gradient descent flow for the data cost is not dependent on the image gradient, which is sensitive to image noise, but on the shape/reflectance estimation.

5.1.2 Gradient Descent Flows for the Normal Deviation Cost and the Shape Area Cost

Similarly as in Jin et al. (2004, 2008), the gradient descent flow for the normal deviation cost $S_t|_{dev}$ (originating from $E_{dev}(\Omega)$) is

$$S_t\big|_{dev} = (-2\tau H + \tau(\nabla \cdot \mathbf{v})), \tag{36}$$

where H is the mean curvature. Also $S_t|_{shape}$ (from $E_{shape}(S)$) is the mean curvature flow as

$$S_t\big|_{shape} = -2\lambda H. \tag{37}$$

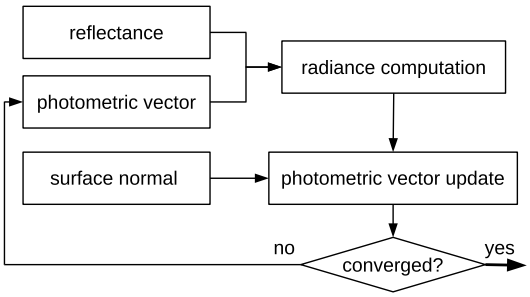


Fig. 4 Photometric unit vector update

5.1.3 *Gradient Descent Flow for the Reflectance Discontinuity Cost*

Due to the complexity of the discontinuity cost function of surface reflectance, it needs more attention to derive the gradient descent flow. By using the derivation in Jin et al. (2003), we get the following equation for surface evolution.

$$S_t|_{refl} = -2\beta\left(\frac{1}{M}m(\rho_d)\eta(R_s) - (m(\rho_s) + \gamma m(\alpha_s))\zeta(R_d)\right). \tag{38}$$

Here,

$$m(\rho_s) = \left(\mathrm{II}\left(\nabla_S\rho_s \times \mathbf{n}\right) + \|\nabla_S\rho_s\|^2 H\right), \tag{39}$$

$$m(\alpha_s) = \left(\mathrm{II}\left(\nabla_S\alpha_s \times \mathbf{n}\right) + \|\nabla_S\alpha_s\|^2 H\right), \tag{40}$$

$$m(\rho_d) = \left(\mathrm{II}\left(\nabla_S\rho_d \times \mathbf{n}\right) + \|\nabla_S\rho_d\|^2 H\right), \tag{41}$$

where $\mathrm{II}(\mathbf{t})$ is the second fundamental form for a tangent vector $\mathbf{t}$ with respect to $\mathbf{n}$.

5.2 Photometric Unit Vector Field Update

The computed gradient descent flows minimize the total cost with respect to given reflectance and $\mathbf{v}$. We then update the photometric unit vector field $\mathbf{v}$ to minimize the total cost with respect to given shape and reflectance. The $\mathbf{v}$ that minimizes the total cost satisfies the equation,

$$\frac{\partial g}{\partial \mathbf{v}} = \left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \mathbf{v}} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3}\right) + (-\tau\mathbf{n}) = 0. \tag{42}$$

Here, $\frac{\partial \bar{I}_i}{\partial \mathbf{v}}$ is given as

$$\frac{\partial \bar{I}_i}{\partial \mathbf{v}} = \sum_{j=1}^{n_{il}} v_{L_{ij}} L_{ij}\left(\rho_d \mathbf{l}_{ij} + \rho_s\alpha_s\left(\mathbf{v}\cdot\mathbf{h}_{ij}\right)^{\alpha_s - 1}\mathbf{h}_{ij}\right). \tag{43}$$

We can update $\mathbf{v}$ by performing gradient descent using the following PDE:

$$\frac{\partial \mathbf{v}}{\partial t} = \left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \mathbf{v}} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3}\right) + (-\tau\mathbf{n}). \tag{44}$$

However, because we have to keep $\|\mathbf{v}\| = 1$, we can not use (44) directly. Since $\mathbf{v} \in \mathbb{S}^2$, $\mathbf{v}$ can be expressed in spherical coordinates as $[\cos\theta_v \sin\phi_v, \sin\theta_v \sin\phi_v, \cos\phi_v]^{\mathrm{T}}$ where θ_v and ϕ_v are the coordinates of $\mathbf{v}$. Therefore, we update θ_v and ϕ_v to update $\mathbf{v}$. As before, the θ_v and ϕ_v that minimize the total cost satisfy the following two equations by the chain rule.

$$\frac{\partial g}{\partial \theta_v} = \frac{\partial g}{\partial \mathbf{v}} \cdot \frac{\partial \mathbf{v}}{\partial \theta_v} = 0, \tag{45}$$

$$\frac{\partial g}{\partial \phi_v} = \frac{\partial g}{\partial \mathbf{v}} \cdot \frac{\partial \mathbf{v}}{\partial \phi_v} = 0. \tag{46}$$

Here, $\frac{\partial \mathbf{v}}{\partial \theta_v}$ and $\frac{\partial \mathbf{v}}{\partial \phi_v}$ are given as

$$\frac{\partial \mathbf{v}}{\partial \theta_v} = \begin{bmatrix} -\sin\theta_v \sin\phi_v \\ \cos\theta_v \sin\phi_v \\ 0 \end{bmatrix}, \qquad \frac{\partial \mathbf{v}}{\partial \phi_v} = \begin{bmatrix} \cos\theta_v \cos\phi_v \\ \sin\theta_v \cos\phi_v \\ -\sin\phi_v \end{bmatrix}. \tag{47}$$

So, we update $\mathbf{v}$ by updating θ_v and ϕ_v by performing gradient descent using the following two PDEs:

$$\frac{\partial \theta_v}{\partial t} = \left(\left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \mathbf{v}} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3}\right) + (-\tau\mathbf{n})\right) \cdot \begin{bmatrix} -\sin\theta_v \sin\phi_v \\ \cos\theta_v \sin\phi_v \\ 0 \end{bmatrix} \tag{48}$$

and

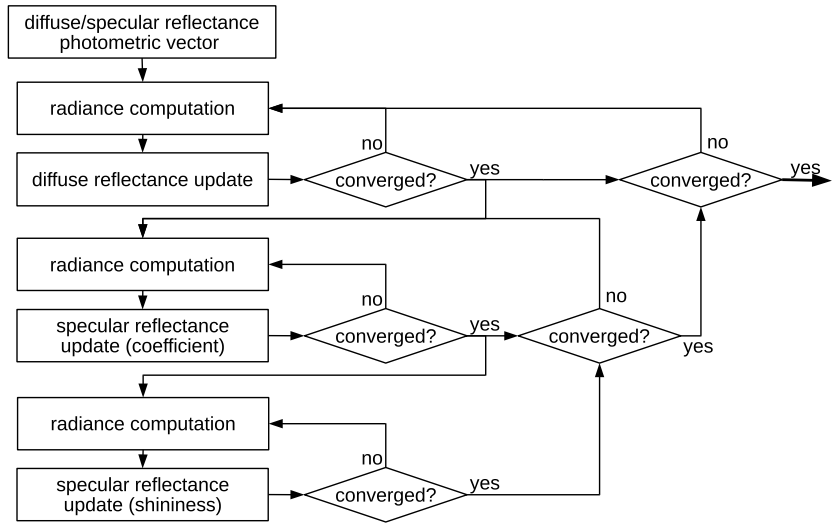
$$\frac{\partial \phi_v}{\partial t} = \left(\left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \mathbf{v}} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3}\right) + (-\tau\mathbf{n})\right) \cdot \begin{bmatrix} \cos\theta_v \cos\phi_v \\ \sin\theta_v \cos\phi_v \\ -\sin\phi_v \end{bmatrix}. \tag{49}$$

5.3 Reflectance Estimation

Here, we estimate R for fixed S and $\mathbf{v}$, still minimizing the total cost function. Since E_{dev} and E_{shape} do not depend on R at all, we seek an optimal R by minimizing $(E_{data}(\Omega) + E_{refl}(R))$. Since it is complex to estimate diffuse and specular reflectance at the same time due to the high coupling between them, we alternatively estimate surface reflectance components one by one while assuming that

Int J Comput Vis (2010) 86: 192–210 201

Fig. 5 Reflectance estimation



the rest are given and fixed. We repeat the procedure until they no longer change. Figure 5 shows the whole scheme we have used for the reflectance estimation. Below, we are detailing the intermediate steps.

5.3.1 Diffuse Reflectance Estimation

For given S and R_s, we estimate ρ_d that minimizes the cost

$$E_{data} + E_{refl} = \int_S \left(\left(-\sum_{i=1}^{n_c} v_S^i \hat{e}_i \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) + \beta \left(1 - \frac{\|\nabla_S \rho_d\|^2}{M} \right) \eta(R_s) \right) d\sigma. \tag{50}$$

Here, ρ_d that minimizes the total cost function will satisfy the Euler-Lagrange equation

$$-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_d} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} + \frac{2\beta}{M} \eta(R_s) \Delta_S \rho_d = 0, \tag{51}$$

where Δ_S denotes the Laplace-Beltrami operator defined on the surface S and $\frac{\partial \bar{I}_i}{\partial \rho_d}$ is given as

$$\frac{\partial \bar{I}_i}{\partial \rho_d} = \sum_{j=1}^{n_{il}} v_{L_{ij}} L_{ij} \left(\mathbf{v} \cdot \mathbf{l}_{ij} \right) + L_{ia}. \tag{52}$$

We solve the PDE by performing gradient descent using the following PDE:

$$\frac{\partial \rho_d}{\partial t} = \left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_d} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) + \left(\frac{2\beta}{M} \eta(R_s) \right) \Delta_S \rho_d. \tag{53}$$

5.3.2 Specular Reflectance Estimation

We then estimate $R_s = (\rho_s, \alpha_s)$ for given S and R_d in the same manner. ρ_s that minimizes the total cost function will satisfy the Euler-Lagrange equation

$$\left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_s} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) - 2\beta \left(\Delta_S \rho_s \right) \zeta \left(\rho_d \right) = 0, \tag{54}$$

where $\frac{\partial \bar{I}_i}{\partial \rho_s}$ is given as

$$\frac{\partial \bar{I}_i}{\partial \rho_s} = \sum_{j=1}^{n_{il}} v_{L_{ij}} L_{ij} \left(\mathbf{v} \cdot \mathbf{h}_{ij} \right)^{\alpha_s}. \tag{55}$$

We again solve the PDE by performing gradient descent using the following PDE to get the solution of (54).

$$\frac{\partial \rho_s}{\partial t} = -\sum_{i=1}^{n_c} \left(v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_s} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) - 2\beta \left(\Delta_S \rho_s \right) \zeta \left(\rho_d \right). \tag{56}$$

α_s is estimated in the same manner by solving the PDE as

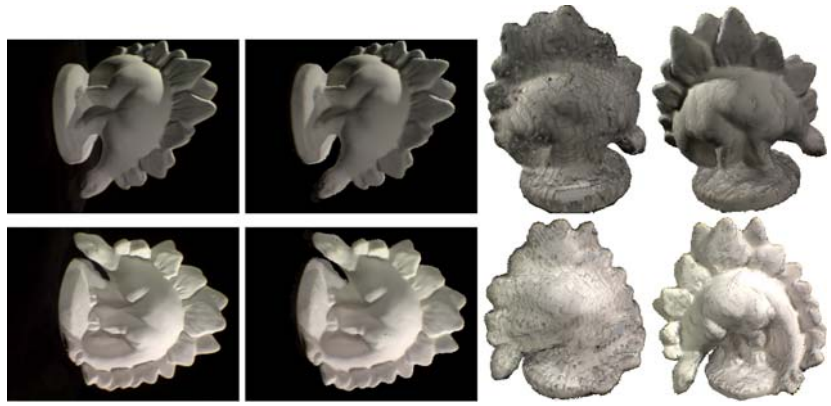
$$\frac{\partial \alpha_s}{\partial t} = -\sum_{i=1}^{n_c} \left(v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \alpha_s} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) - 2\beta\gamma \left(\Delta_S \alpha_s \right) \zeta \left(\rho_d \right), \tag{57}$$

where $\frac{\partial \bar{I}_i}{\partial \alpha_s}$ is given as

$$\frac{\partial \bar{I}_i}{\partial \alpha_s} = \sum_{j=1}^{n_{il}} v_{L_{ij}} L_{ij} \rho_s \left(\mathbf{v} \cdot \mathbf{h}_{ij} \right)^{\alpha_s} \ln \left(\mathbf{v} \cdot \mathbf{h}_{ij} \right). \tag{58}$$

5.3.3 Case of a Single-Material Surface

When dealing with a surface that has uniform specular reflectance R_s, it is possible to set $\rho_s(\mathbf{X}) = \rho_s$ and $\alpha_s(\mathbf{X}) = \alpha_s$



(a) input images (b) synthesized images (c) initial shape (d) estimated shape

Fig. 6 Result for the "dino" image set (16 images)—Lambertian surface case (static illumination and varying viewpoint)

for all surface points. In this case, the discontinuity cost function of surface reflectance, $E_{refl}(R)$, can be excluded because $f(\mathbf{X})$ in (21) is zero everywhere on the surface. Hence, the gradient descent flow is then given by

$$S_t = \left(S_t|_{data} + S_t|_{dev} + S_t|_{shape}\right), \tag{59}$$

and the PDE used for the estimation of ρ_d, (53), is simplified to

$$\frac{\partial \rho_d}{\partial t} = -\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_d} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3}. \tag{60}$$

In addition, ρ_s and α_s are computed by performing gradient descent using the following PDEs.

$$\frac{\partial \rho_s}{\partial t} = \int_S \left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \rho_s} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) d\sigma, \tag{61}$$

$$\frac{\partial \alpha_s}{\partial t} = \int_S \left(-\sum_{i=1}^{n_c} v_S^i \partial_2 \hat{e}_i \frac{\partial \bar{I}_i}{\partial \alpha_s} \frac{\mathbf{d}_i \cdot \mathbf{n}}{{z_i}^3} \right) d\sigma. \tag{62}$$

6 Experiments

6.1 Implementation

We have implemented the gradient descent surface evolution in the level set framework in which the topological changes of surfaces are handled automatically (Osher and Sethian 1988; Sethian 1999; Osher and Fedkiw 2002). The proposed method starts with the visual hull obtained by rough silhouette images to reduce computational time and to avoid local minima. We also adopt a multi-scale strategy. 640×480 or 800×600 images were used as inputs and the simple L^2-norm was used to compute the image similarity, e. The camera and light visibility were computed using OpenGL z-buffering.[7] In all experiments, we detected saturated pixels by thresholding the intensity against $I_{th} = 253$, and ignored them in all further computations.

For synthetic data sets, the estimated shape is quantitatively evaluated in terms of accuracy and completeness as in Seitz et al. (2006). We used 95% for accuracy and the 1.0 mm error for completeness. For easy comprehension, the size of a target object is normalized so that it is smaller than [100 mm 100 mm 100 mm]. Here, beside the shape evaluation, we also evaluated the estimated reflectance in the same manner. For each point on an estimated surface, we found the nearest point on the true surface and compute the distance and reflectance differences, and vice versa. In addition, we computed the average difference between input images and synthesized images as

$$e_{image} = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{1}{A} \int_{\pi_i} \left\| \left(I_i(\mathbf{x}) - \bar{I}_i(\mathbf{x}) \right) \right\| d\sigma_i, \tag{63}$$

where $A = \int_{\pi_i} d\sigma_i$.

6.2 Experimental Results

Due to the generality of the proposed method, it can be applied to various types of image sets with different camera/light configurations. Here, knowledge of illumination allows to factorize radiance into reflectance and geometry. In practice, depending on the scenario, that knowledge may not be required, e.g. for recovering shape and radiance of Lambertian surfaces with static illumination. In other words, when images of Lambertian surfaces are taken under static illumination, the proposed method can be applied even without lighting information, assuming that there is only an ambient illumination. In this case, we do not need to take care of surface/photometric normals and only $E_{data}(\Omega)$ and

[7] Light visibility is computed by using virtual cameras located at the positions of light sources.

Int J Comput Vis (2010) 86: 192–210 203

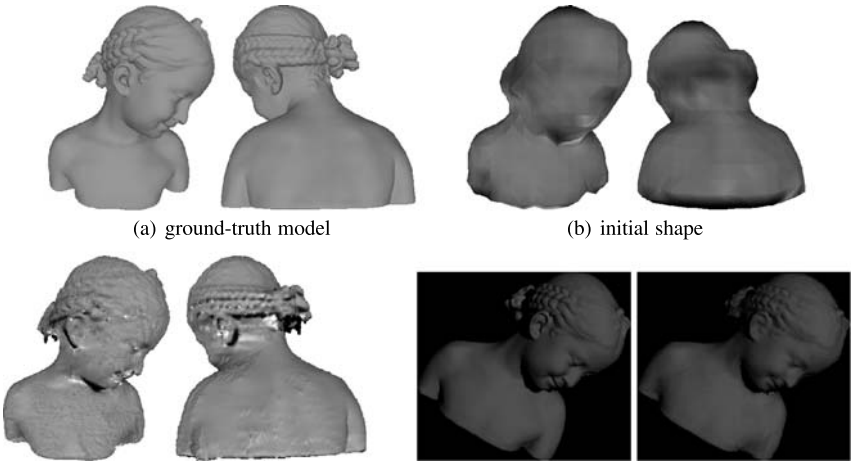

(a) ground-truth model (b) initial shape
(c) estimated model (d) input image vs. synthesized image

Fig. 7 Result for the "bimba" image set (18 images)—textureless Lambertian surface case (varying illumination and viewpoint). 95% accuracy (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}$) = (2.16 mm, 0.093, 0.093, 0.093), 1.0 mm completeness (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}$) = (82.63%, 0.104, 0.104, 0.104), $e_{image} = 1.44$

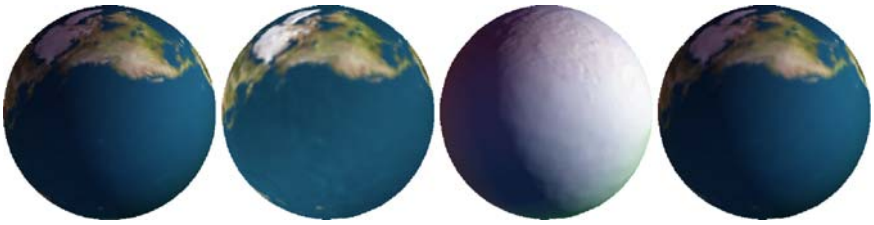

(a) input image (b) estimated reflectance (c) estimated shading (d) synthesized image

Fig. 8 Result for the "sphere" image set (32 images)—textured Lambertian surface case (static illumination and varying viewpoint). 95% accuracy (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}$) = (1.06 mm, 0.025, 0.019, 0.017), 1.0 mm completeness (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}$) = (99.74%, 0.023, 0.017, 0.016), $e_{image} = 0.60$

$E_{shape}(S)$ are computed. The intensity conservation assumption (ICA) is valid in all images and the proposed method works much like the conventional multiview stereo methods and estimates the shape and radiance of Lambertian surfaces. Figure 6 shows the result for the dino image set (Seitz et al. 2006), for which no lighting information is required. The proposed method successfully recovers the shape as well as the radiance.

The proposed method can also be applied to images taken under varying illumination. Results using images of textureless/textured Lambertian surfaces are shown in Figs. 7 to 12. Figure 7 shows the ground-truth shape of the "bimba" image set (18 images) of a textureless object, and the estimation result. The surface has uniform diffuse reflectance and input images were taken under different illuminations. In this case, the proposed method works as a multiview photometric stereo method and recovers the shape and the diffuse reflectance of each surface point. Here, black points in the estimated model correspond to points that were not visible from any camera and/or any light source.

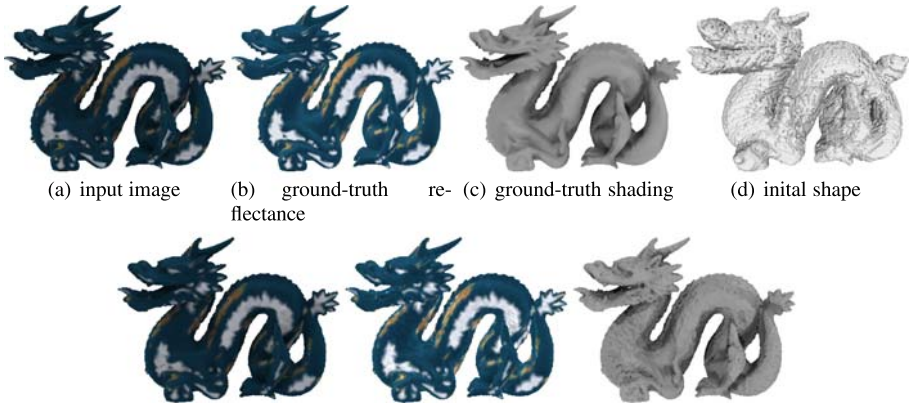
Figure 8 shows one of 32 textured input images and the synthesized image generated using the estimated shape (i.e., shading) and reflectance. Based on this result, we can also synthesize images of the scene for different lighting conditions, as shown in Fig. 9. Results for a more complex object

(a) original input image (b) synthesized image

Fig. 9 Image synthesis—an image of the same scene, for different lighting conditions

are shown in Figs. 10 and 11. The images synthesized using the estimation closely resemble input images while the shading and the reflectance are successfully separated. Furthermore, it is possible to synthesize images under different lighting conditions, even from different viewpoints. The proposed method also recovers concave parts well as shown in Fig. 12.

We then applied our method to the images of textureless/textured non-Lambertian surfaces showing specular reflection. Note that, unlike previous methods (Birkbeck et al. 2006; Hernández Esteban et al. 2008), we do not use any thresholding to filter out specular highlight pixels. The result for the smoothed "bimba" data set is shown in Fig.



(a) input image (b) ground-truth reflectance (c) ground-truth shading (d) inital shape

(e) synthesized image (f) estimated reflectance (g) estimated shading

Fig. 10 Result for the "dragon" image set (32 images)—textured Lambertian surface case (static illumination and varying viewpoint). 95% accuracy (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}) =$ (1.28 mm, 0.090, 0.073, 0.066), 1.0 mm completeness (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}) =$ (97.11%, 0.064, 0.056, 0.052), $e_{image} = 1.25$

(a) ground-truth rendered with a different lighting and from a new viewpoint (b) ground-truth reflectance (c) ground-truth shading

(d) reconstructed scene relighted as in (a) and viewed from the same point of view as (a) (e) estimated reflectance (f) estimated shading

Fig. 11 Synthesized result for different lighting conditions and viewed from a viewpoint that is different from all input viewpoints. A comparison with the ground-truth is possible because this is synthetic data

(a) ground-truth (b) close-up view (c) close-up view of the estimated result

Fig. 12 Close-up view of the concave part of the "dragon" model

13. In this case, the surface has uniform diffuse/specular reflectance and each image was taken under a different illumination. Here, we used the method described in Sect. 5.3.3 to estimate the specular reflectance. Although there is high-frequency noise in the estimated shape, the proposed method estimates the specular reflectance well—the ground-truth specular reflectance is $(\rho_s = 0.7, \alpha_s = 50)$ while the estimated one is $(\rho_s = 0.61, \alpha_s = 41.8)$.[8]

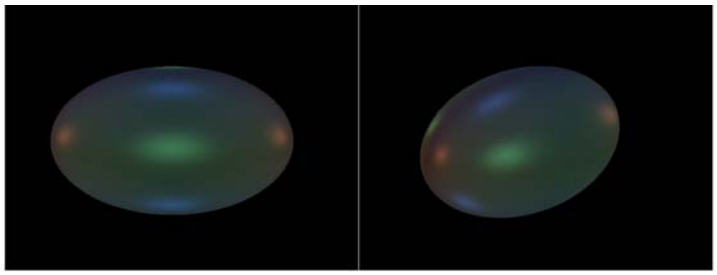
[8]Note that small errors in estimated surface normals can cause large errors in specular reflectance because of its sensitivity to the surface normal. For instance, $0.7 \times (0.98)^{50} (= 0.255) \approx 0.61 \times (0.979)^{41.8} (= 0.251)$.

Int J Comput Vis (2010) 86: 192–210 205

(a) ground-truth model (b) estimated shape (c) diffuse image (d) specular image (e) synthesized image

Fig. 13 Result for the smoothed "bimba" image set (36 images)—textureless non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint). 95% accuracy (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}, \rho_s, \alpha_s) = (0.33\text{ mm}, 0.047, 0.040, 0.032, 0.095, 8.248)$, 1.0 mm completeness (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}, \rho_s, \alpha_s) = (100\%, 0.048, 0.041, 0.032, 0.095, 8.248)$, $e_{image} = 1.63$

Fig. 14 Result comparison using the "ellipse" image set (16 images)—textureless non-Lambertian surface case (uniform specular reflectance, static illumination and varying viewpoint)



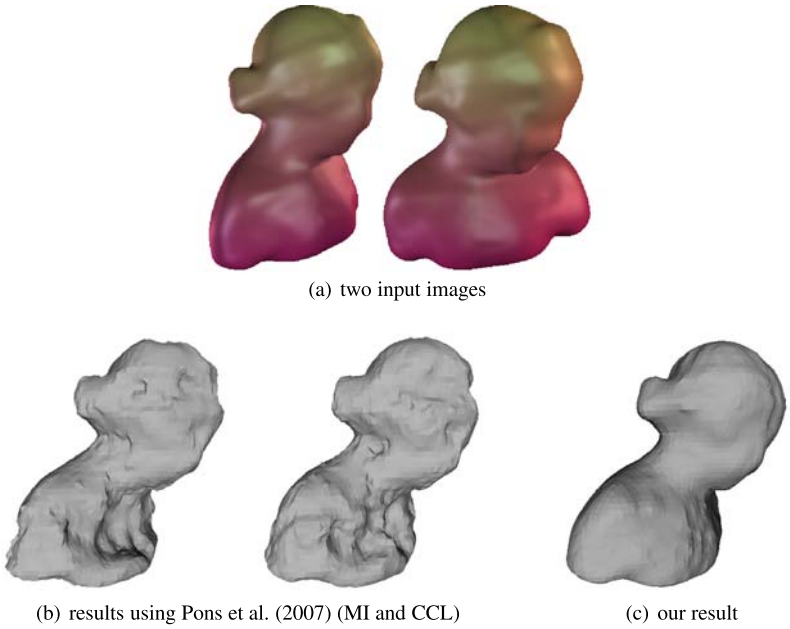
(a) two input images

(b) results using Pons et al. (2007) (MI and CCL) (c) our result

Note that most previous methods do not work for image sets taken under varying illumination and, moreover, they have difficulties to deal with specular reflection even if the images are taken under static illumination. For example, Figs. 14 and 15 show some results obtained by the method of Pons et al. (2007) and our result for comparison. We ran the original code provided by the authors many times while changing parameters and used mutual information (MI) and cross correlation (CCL) as similarity measures to get the best results under specular reflection. As shown in Figs. 14 and 15, the method of Pons et al. (2007) fails to get a good shape even when the shape is very simple, while our method estimates it accurately. Also, with such images, given the large proportion of overbright surface parts, it seems intuitive that the strategy chosen by Birkbeck et al. (2006) and Hernández Esteban et al. (2008) (who consider bright pixels as outliers) might return less accurate results, because it removes too much information.

We also used real image sets of textured glossy objects, which were taken by using fixed cameras/light sources, while rotating the objects as in Birkbeck et al. (2006), Hernández Esteban et al. (2008)—in this case, each image has a different illumination and observes specular reflections. The light position and color were measured using a white sphere placed in the scene. Figure 16 shows one image among 59 input images, the initial shape obtained using silhouettes, and the final result. Here, we simply assumed a single-material surface (i.e. uniform specular reflectance, but varying albedo). More results using real image sets are shown in Figs. 17 to 18. $(72 \times 72 \times 72)$ grids were used for the "saddog" and "duck" image sets and $(64 \times 64 \times 64)$ grids for "bunny". Although sparse grid volumes were used, the proposed method successfully estimated the shape of the glossy object even under specular reflection, while estimating the latter. Here, we can see that, although the estimated specular reflectance may not be highly accurate because of the inaccuracy of lighting calibration, saturation, and unmodeled photometric phenomena such as interreflections that often occur on glossy surfaces, it really helps to recover the shape well.

Finally, we applied our method to the most general case—images of textured non-Lambertian surfaces with

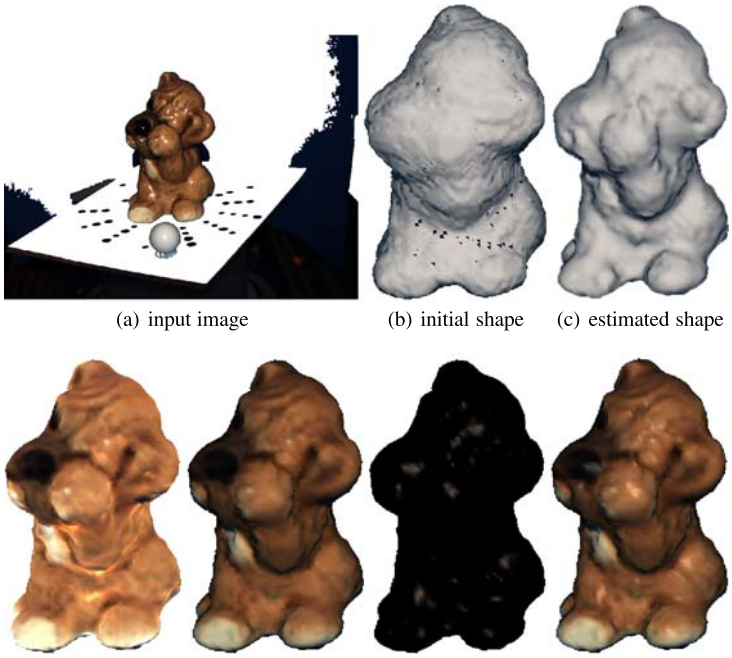


(a) two input images

(b) results using Pons et al. (2007) (MI and CCL)

(c) our result

Fig. 15 Result comparison using the smoothed "bimba" image set (16 images)—textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint)



(a) input image (b) initial shape (c) estimated shape

(d) diffuse reflectance (e) diffuse image (f) specular image (g) synthesized image

Fig. 16 Result for the "saddog" image set (59 images)—textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint). $e_{image} = 2.45$

spatially varying diffuse and specular reflectance and shininess, cf. Fig. 19. Input images were generated under static illumination (with multiple light sources) while changing the viewpoint. Figure 19 shows one image among 36 input images, one ground-truth diffuse image, one ground-truth specular image, ground-truth shading, and our results. $(64 \times 125 \times 64)$ grids were used in this case. We can see that the proposed method yields plausible specular/diffuse images and shape. However, there is high-frequency noise in the estimated shape. Moreover, the error in reflectance estimation is rather larger compared to the previous cases because of sparse specular reflection observation. This result shows that, although the proposed discontinuity cost function of surface reflectance helps to infer the specular reflectance of all points with sparse specular reflection observation, reliably estimating specular reflectance for all surface points is still difficult unless there are enough observation of specular reflections for every surface point.

Int J Comput Vis (2010) 86: 192–210 207

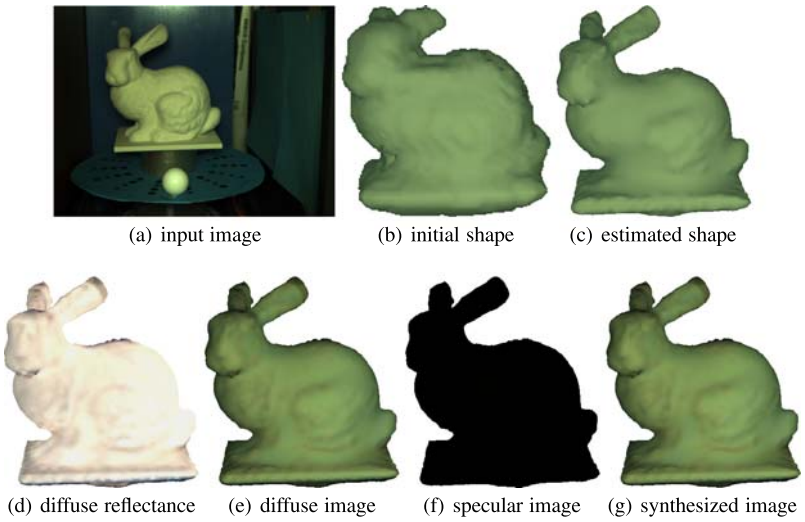

(a) input image (b) initial shape (c) estimated shape
(d) diffuse reflectance (e) diffuse image (f) specular image (g) synthesized image

Fig. 17 Result for the "bunny" image set (26 images)—textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint). $e_{image} = 5.34$

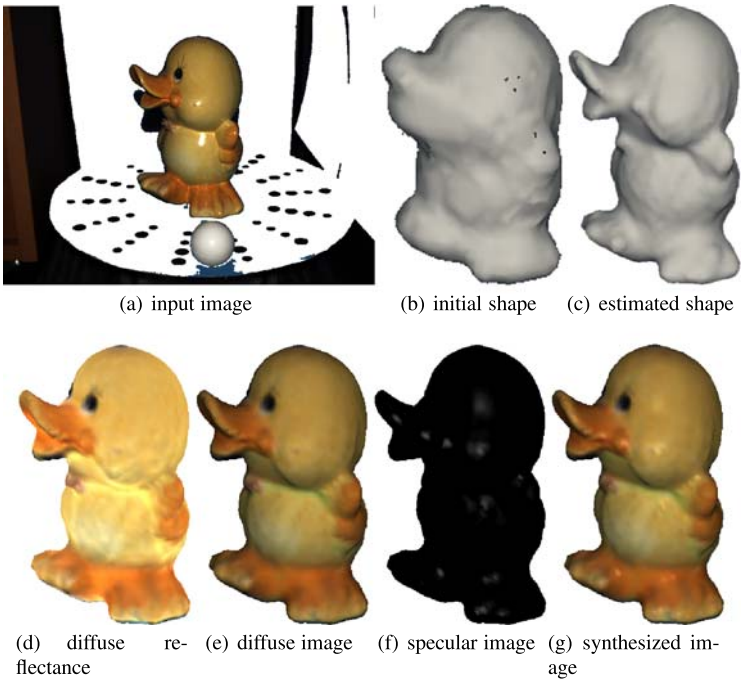

(a) input image (b) initial shape (c) estimated shape
(d) diffuse reflectance (e) diffuse image (f) specular image (g) synthesized image

Fig. 18 Result for the "duck" image set (28 images)—textured non-Lambertian surface case (uniform specular reflectance, varying illumination and viewpoint). $e_{image} = 2.79$

7 Discussion and Further Work

When considering non-Lambertian surfaces under varying illumination, the core question is how to deal with specular reflection in shape estimation. As mentioned in the introduction, one common way is to decouple shape and reflectance estimation using reflectance invariants or a specular-free image representation as in Yoon and Kweon (2006), Mallick et al. (2005), Zickler et al. (2008). In this approach, input images are transformed so that they are free from specular reflection and the resultant images are then used for surface shape recovery. Reflectance can then be recovered using the estimated shape. This approach might be less computationally intensive and more robust/stable than the proposed method because it is not required to consider the complex specular reflection during shape estimation. However, it is strongly limited by the specific lighting configuration—the transformation is valid only when the illumination conditions have specific properties. While it is also useful to detect specular highlights first and to treat them as outliers as in Hernández Esteban et al. (2008), Birkbeck et al. (2006), detecting specular highlights is a hard problem in itself.

The proposed method uses the Blinn-Phong shading model to describe specular reflectance, but it is also possible to use any other parametric reflectance model. The parametric reflectance model is directly incorporated in the problem formulation. As a result, the proposed method can be applied to various data sets, and is not limited by the specific lighting

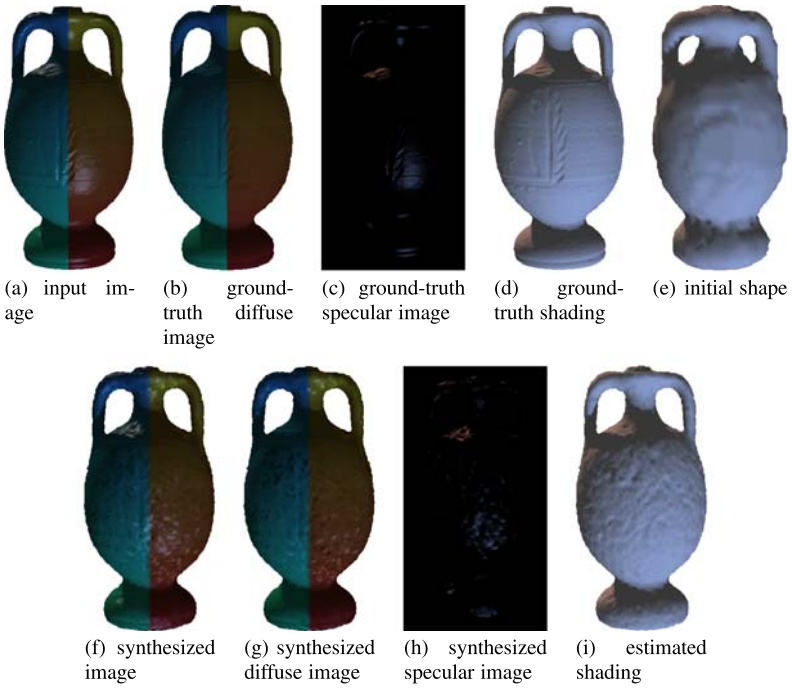


Fig. 19 Result for the "amphora" image set (36 images)—textured non-Lambertian surface case (spatially varying specular reflectance, static illumination, and varying viewpoint). 95% accuracy (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}, \rho_s, \alpha_s) = (0.59$ mm, 0.041, 0.047, 0.042, 0.226, 12.69), 1.0 mm completeness (shape, $\rho_{dr}, \rho_{dg}, \rho_{db}, \rho_s, \alpha_s) = (89.73\%, 0.042, 0.047, 0.042, 0.226, 12.65)$, $e_{image} = 1.99$

configuration. This is one of the main contributions of the proposed method. However, some recent studies (Ngan et al. 2005; Stark et al. 2005) have shown conclusively that, although parametric reflectance models are widely used in the literature, they are often unable to capture important visual effects. This is especially true for specular reflectance. As a result, the proposed method may produce rather inaccurate results for real images.

Moreover, the proposed method can become less stable and less accurate as the scenario gets more complex. For example, it works very well for diffuse images taken under static illumination, which is the simplest case, because we do not need to consider the surface normal and the photometric normal of each point. However, when dealing with non-Lambertian surfaces, the estimates of shape and reflectance are rather less accurate. In some aspects, this is natural because the proposed method deals with many spatially varying unknowns and estimates them alternatively. As described, the proposed method consists of many sub-parts, which also have alternative loops in them. It suffers from local minima and the alternative scheme sometimes fails in practice. In addition, for non-Lambertian cases, the proposed method may produce inaccurate results because specular reflection can be extremely sensitive to the surface normal depending on the surface shininess. On the other hand, when specular reflectance occurs broadly and its magnitude is small compared to that of diffuse reflectance, the problem can be ambiguous and the proposed method may return poor results because it is hard to distinguish the specular reflectance from the diffuse one in this case. The accuracy of specular reflectance estimation depends on the number of specular reflection observations as well. However, by combining multiple cues, the dependency on the initial shape can be reduced.

Two other difficulties of the proposed method are computational time and huge memory requirement. In fact, the proposed method is more computationally expensive than shape and reflectance decoupling approaches. All sub-loops should converge at each iteration as shown in Figs. 1, 2, 4, and 5, and the shape can not evolve much at each iteration because of the stability. Therefore, the proposed method takes from a few hours to a few days according to the image sets and the initial conditions—the computational time is in proportion to the resolution of the grid used in the level set framework, the number of input images, and the number of light sources. In addition, contrary to previous works, we rigorously consider camera and light visibility and self-occlusion. This incurs huge memory requirements, precluding the use of a dense level set grid. Therefore, we have some difficulties to deal with complex shaped objects in experiments, which is why we used the smoothed bimba data set in Fig. 13. In addition, because of the sparseness of the level set grid, the initial shape given by rough silhouette images is not accurate, so the initially computed specular reflectance is not close to the ground truth. However, even when we use a small number of images, we estimate the specular reflectance at each iteration and this improves the accuracy of the method.

In other respects, the proposed method needs several user-specified parameters such as τ, λ, and β in (27). These parameters control the contribution of the individual cost functions, according to the types of image sets. As a result,

the overall performance may be biased towards specific priors according to these parameters.

As a future work, we would like to develop faster, more stable, and more accurate schemes to overcome these limitations. Also, there is some other room for improvement. For example, we used the simple L^2-norm for computing e, while it is possible to use other global/robust measures such as cross correlation or mutual information as in Pons et al. (2007). It is also possible to change the inner product structure as proposed by Charpiat et al. (2007). In addition, in this work, we adopted the Blinn-Phong model to describe specular reflectance, but it should be relevant to adopt a more realistic model.

8 Conclusion

In this paper, we have presented a variational method that recovers both the shape and the reflectance of surfaces using multiple images, assuming that illumination conditions and camera calibration are known in advance. Scene recovery is achieved by minimizing a global cost functional by alternation. As a result, the proposed method produces a complete description of scene surfaces.

Contrary to previous works that consider specific scenarios, our method can be applied indiscriminately to a number of classical scenarios—it naturally fuses and exploits several important cues (silhouettes, stereo, and shading) and allows to deal with most of the classical 3D reconstruction scenarios such as stereo vision, (multi-view) photometric stereo, and multiview shape from shading. In addition, our method can deal with non-Lambertian surfaces showing strong specular reflection, which is difficult even in some other state of the art methods using complex similarity measures.

Acknowledgements We would like to thank Neil Birkbeck for providing real data sets for our experiments and Jean-Philippe Pons and the CERTIS Lab for providing their C++ level-set library and their multi-view stereo vision code.

References

Bhat, D. N., & Nayar, S. K. (1998). Stereo and specular reflection. *International Journal of Computer Vision*, *26*(2), 91–106.

Birkbeck, N., Cobzas, D., Sturm, P., & Jägersand, M. (2006). Variational shape and reflectance estimation under changing light and viewpoints. In *European conference on computer vision* (Vol. 1, pp. 536–549).

Blinn, J. F. (1997). Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77: Proceedings of the 4th annual conference on computer graphics and interactive techniques* (pp. 192–198). New York: ACM. http://doi.acm.org/10.1145/563858.563893.

Charpiat, G., Maurel, P., Pons, J. P., Keriven, R., & Faugeras, O. (2007). Generalized gradients: Priors on minimization flows. *International Journal of Computer Vision*, *73*(3), 325–344.

Faugeras, O., & Keriven, R. (1998). Variational-principles, surface evolution, PDEs, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, *7*(3), 336–344.

Gargallo, P. (2008). *Contributions to the Bayesian approach to multiview stereo*. PhD thesis, Institut National Polytechique de Grenoble, France.

Gargallo, P., Prados, E., & Sturm, P. (2007). Minimizing the reprojection error in surface reconstruction from images. In *IEEE international conference on computer vision*.

Georghiades, A. S. (2003). Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In *IEEE international conference on computer vision* (Vol. 02, pp. 816–823).

Goesele, M., Curless, B., & Seitz, S. M. (2006). Multi-view stereo revisited. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 2402–2409).

Goldman, D. B., Curless, B., Hertzmann, A., & Seitz, S. M. (2005). Shape and spatially-varying BRDFs from photometric stereo. In *IEEE international conference on computer vision* (pp. 341–348).

Hernández Esteban, C., Vogiatzis, G., & Cipolla, R. (2008). Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(3), 548–554.

Hertzmann, A., & Seitz, S. M. (2005). Example-based photometric stereo: Shape reconstruction with general, varying BRDFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1254–1264.

Jin, H., Yezzi, A., & Soatto, S. (2002). Variational multiframe stereo in the presence of specular reflections. In *International symposium on 3D data processing visualization and transmission* (pp. 626–630).

Jin, H., Yezzi, A. J., Tsai, Y. H., Cheng, L. T., & Soatto, S. (2003). Estimation of 3D surface shape and smooth radiance from 2D images: A level set approach. *Journal of Scientific Computing*, *19*(1–3), 267–292.

Jin, H., Cremers, D., Yezzi, A. J., & Soatto, S. (2004). Shedding light on stereoscopic segmentation. In *IEEE conference on computer vision and pattern recognition* (Vol. 1, pp. 36–42).

Jin, H., Soatto, S., & Yezzi, A. J. (2005). Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, *63*(3), 175–189.

Jin, H., Cremers, D., Wang, D., Prados, E., Yezzi, A., & Soatto, S. (2008). 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, *76*(3).

Kim, J., Kolmogorov, V., & R Zabih, R. (2003). Visual correspondence using energy minimization and mutual information. In *IEEE international conference on computer vision* (pp. 1033–1040).

Kolev, K., Klodt, M., Brox, T., & Cremers, D. (2007a). Propagated photoconsistency and convexity in variational multiview 3d reconstruction. In *Workshop on photometric analysis for computer vision*, Rio de Janeiro, Brazil.

Kolev, K., Klodt, M., Brox, T., Esedoglu, S., & Cremers, D. (2007b). Continuous global optimization in multiview 3d reconstruction. In *LNCS: Vol. 4679. Energy minimization methods in computer vision and pattern recognition (EMMCVPR)*, Ezhou, China (pp. 441–452). Berlin: Springer.

Lee, H. C., Breneman, E. J., & Schulte, C. P. (1990). Modeling light reflection for computer color vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(4), 402–409.

Lu, J., & Little, J. (1995). Reflectance function estimation and shape recovery from image sequence of a rotating object. In *IEEE international conference on computer vision* (pp. 80–86).

Mallick, S. P., Zickler, T., Kriegman, D. J., & Belhumeur, P. N. (2005). Beyond Lambert: Reconstructing specular surfaces using color. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 619–626).

Ngan, A., Durand, F., & Matusik, W. (2005). Experimental analysis of BRDF models. In *Eurographics symposium on rendering* (pp. 117–126).

Osher, S., & Fedkiw, R. (2002). *The level set method and dynamic implicit surfaces*. Berlin: Springer.

Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, *79*, 12–49. citeseer.ist.psu.edu/osher88fronts.html.

Paris, S., Sillion, F. X., & Quan, L. (2006). A surface reconstruction method using global graph cut optimization. *International Journal of Computer Vision*, *66*(2), 141–161.

Pons, J. P., Keriven, R., & Faugeras, O. (2005). Modelling dynamic scenes by registering multi-view image sequences. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 822–827).

Pons, J. P., Keriven, R., & Faugeras, O. (2007). Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *International Journal of Computer Vision*, *72*(2), 179–193.

Powell, M. W., Sarkar, S., & Goldgof, D. (2001). A simple strategy for calibrating the geometry of light sources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(9), 1022–1027.

Schultz, H. (1994). Retrieving shape information from multiple images of a specular surface. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(2), 195–201. http://dx.doi.org/10.1109/34.273732.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE conference on computer vision and pattern recognition* (pp. 519–528).

Sethian, J. (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials sciences*. *Cambridge monograph on applied and computational mathematics*. Cambridge: Cambridge University Press.

Snow, D., Viola, P., & Zabih, R. (2000). Exact voxel occupancy with graph cuts. In *IEEE conference on computer vision and pattern recognition* (Vol. 1, pp. 345–352).

Soatto, S., Yezzi, A. J., & Jin, H. (2003). Tales of shape and radiance in multi-view stereo. In *IEEE international conference on computer vision* (pp. 974–981).

Solem, J. E., & Overgaard, N. C. (2005). A geometric formulation of gradient descent for variational problems with moving surfaces. In *Scale-space* (pp. 419–430).

Stark, M., Arvo, J., & Smits, B. (2005). Barycentric parameterizations for isotropic BRDFs. *IEEE Transactions on Visualization and Computer Graphics*, *11*(2), 126–138.

Tran, S., & Davis, L. (2006). 3d surface reconstruction using graph cuts with surface constraints. In *European conference on computer vision* (Vol. 2, pp. 219–231).

Vogiatzis, G., Favaro, P., & Cipolla, R. (2005). Using frontier points to recover shape, reflectance and illumination. In *IEEE international conference on computer vision* (Vol. 1, pp. 228–235).

Vogiatzis, G., Hernández Esteban, C., Torr, P. H. S., & Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2241–2246.

Yang, R., Pollefeys, M., & Welch, G. (2003). Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure. In *IEEE international conference on computer vision* (pp. 576–583).

Yezzi, A., & Soatto, S. (2003). Stereoscopic segmentation. *International Journal of Computer Vision*, *53*(1), 31–43.

Yoon, K. J., & Kweon, I. S. (2006). Correspondence search in the presence of specular highlights using specular-free two-band images. In *Asian conference on computer vision* (pp. 761–770).

Yu, T., Xu, N., & Ahuja, N. (2004). Recovering shape and reflectance model of non-lambertian objects from multiple views. In: *IEEE conference on computer vision and pattern recognition* (pp. 226–233).

Yu, T., Xu, N., & Ahuja, N. (2007). Shape and view independent reflectance map from multiple views. *International Journal of Computer Vision*, *73*(2), 123–138.

Zach, C., Sormann, M., & Karner, K. (2006). High-performance multi-view reconstruction. In *3DPVT* (pp. 113–120). IEEE Computer Society.

Zhou, W., & Kambhamettu, C. (2002). Estimation of illuminant direction and intensity of multiple light sources. In *European conference on computer vision* (pp. 206–220).

Zickler, T. (2006). Reciprocal image features for uncalibrated Helmholtz stereopsis. In *IEEE conference on computer vision and pattern recognition* (pp. 1801–1808).

Zickler, T., Belhumeur, P. N., & Kriegman, D. J. (2002). Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, *49*(2–3), 215–227.

Zickler, T., Mallick, S. P., Kriegman, D. J., & Belhumeur, P. (2008). Color subspaces as photometric invariants. *International Journal of Computer Vision*, *79*(1), 13–30.

Gradient Flows for Optimizing Triangular Mesh-based Surfaces: Applications to 3D Reconstruction Problems dealing with Visibility*

Amaël Delaunoy Emmanuel Prados

Abstract

This article tackles the problem of using variational methods for evolving 3D deformable surfaces. We give an overview of gradient descent flows when the shape is represented by a triangular mesh-based surface, and we detail the gradients of two generic energy functionals which embody a number of energies used in mesh processing and computer vision. In particular, we show how to rigorously account for visibility in the surface optimization process. We present different applications including 3D reconstruction from multiple views for which the visibility is fundamental. The gradient correctly takes into account the visibility changes that occur when a surface moves; this forces the contours generated by the reconstructed surface to match with the apparent contours in the input images.

1 Introduction

Variational methods are commonly used in computer vision and computer graphics to compute, improve and process surface interfaces. Such approaches consist of defining an energy whose minimum is reached by the surface of the object of interest. In particular, this framework has been widely used in 3D reconstruction problems, see for example [5, 13, 15, 22, 23, 24, 26, 29, 35, 40, 41, 46, 47, 49]. In mesh processing, geometric flows have been extensively used in different applications such as texture synthesis [2], mesh denoising [9, 30] and shape matching [12].

This article focuses on the optimization of *2D surfaces of* $\mathbb{R}^3$ represented by *triangle meshes*, via gradient descent methods. We rigorously establish and detail the *gradient flows* of some *generic* energies which encompass a large number of energies used in computer vision and for which the normal to the surface and the *visibility* appear in their formulation. We demonstrate the interest of this contribution by illustrating it via several applications, in particular applications in 3D reconstruction from multiple calibrated cameras.

1.1 Considered Energies: from Weighted Area Functionals to Functionals that account for Visibility

In this paper, we first consider the following generic energy:

$$E(\mathcal{S}) = \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\mathrm{d}s \,, \tag{1}$$

where $\mathcal{S}$ is a $2D$ surface embedded into $\mathbb{R}^3$. Here $g : \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$ is a scalar function defined on the surface that eventually depends on the normal $\mathbf{n}$ to the surface $\mathcal{S}$; $\mathbb{S}$ being the unit sphere of $\mathbb{R}^3$. $\mathrm{d}s$ is the element of area of the surface. This generic energy is very classical and is called a weighted area functional. It has already been studied in the literature in the continuous framework, see in particular [16, 17, 42]. A number of energies proposed in the computer vision, image processing and mesh processing literature are particular cases of this energy.

In this paper we are also considering more complex families of energies. Generally in inverse problems, all rests on a priori knowledge (models, regularizations etc)

*Research was supported by the Agence Nationale pour la Recherche within the Flamenco project (Grant ANR-06-MDCA-007).

and the data fidelity. A common solution to inverse problems is provided by minimizing some criteria which compares the real input data to some synthetic data generated by the models. Also, to be complete, the comparisons must be done on the data space. In particular, as explained in [15, 39], in 3D reconstruction problems from image data, a natural solution would be a surface such that the images generated from the model are more similar to the observed images (i.e. the data). This naturally leads to formulate the problem as the minimization of an error measure between the observed and predicted values of pixels, carried out over all pixels in all input images. This thus brings us to minimize some energy of the form [15, 39]:

$$E(\mathcal{S}) = \int_{\mathcal{I}} g\big(\, \pi_{\mathcal{S}}^{-1}(\mathbf{p}),\, \mathbf{n}(\pi_{\mathcal{S}}^{-1}(\mathbf{p}))\,\big)\, \mathrm{d}\mathbf{p}\,, \qquad (2)$$

where $\mathcal{I}$ is the set on which data is defined (the set of all pixels in the image, in the case of the 3D reconstruction problems from images), and $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ is the point of the surface corresponding to $\mathbf{p}$ (the surface point which is viewed in pixel $\mathbf{p}$, in the 3D reconstruction problems from images). In the case where $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ does not reproject onto the surface $\mathcal{S}$, $\pi_{\mathcal{S}}^{-1}(\mathbf{p})$ is a point on the background B. $\mathrm{d}\mathbf{p}$ is the area measure on the sensor. g gives the error measure for the data point $\mathbf{p}$. Such functionals are generally called reprojection errors. One of the major properties in image formation, also one of the major problems in computer vision, is that only visible (i.e. unoccluded) elements are present in the image. Functionals (2) can then be rewritten as an integral over the surface (instead of the image) by counting only the visible points [34, 39, 47]. This gives, by a simple change of variables:

$$E(\mathcal{S}) = \int_{\mathcal{S}\cup B} \overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\,. \qquad (3)$$

This involves adapting the measure on the surface [39] and the expression of $\overline{\mathbf{g}}$ which directly depends on the projection model being used. In most cases, e.g. for orthographic, perspective or linear pushbroom cameras, we have $\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n}) k(\mathbf{x}) \mathbf{d}(\mathbf{x})$, where $k(\mathbf{x})$ is a specific scalar function of $\mathbf{x}$ and $\mathrm{d}(\mathbf{x})$ is the projection vector of $\mathbf{x}$ according to the camera. For example, for a perfect pinhole camera model, the adequate $\overline{\mathbf{g}}$ is given by $\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n}) \frac{1}{\mathbf{x}_z^3} \mathbf{x}$, see [34, 39, 47]. For a linear pushbroom camera, we have $\overline{\mathbf{g}}(\mathbf{x}, \mathbf{n}) = -g(\mathbf{x}, \mathbf{n}) \frac{1}{v \cdot \mathbf{d}(\mathbf{x})\, \mathrm{d}_z(\mathbf{x})} \mathbf{d}(\mathbf{x})$, where $\mathbf{d}(\mathbf{x})$ is the vector joining $\mathbf{x}$ and the optical center of the sensor at corresponding time; v is a vector depending on the speed of the satellite, see [18]. In (3), B is the surface behind $\mathcal{S}$ that corresponds to the background (i.e. the points on the data set which do not correspond to any point of the surface model of the object of interest). $\nu_{\mathcal{S}}(\mathbf{x})$ is the visibility function $\nu_{\mathcal{S}} : \mathbb{R}^3 \mapsto \mathbb{R}^3$ such that:

$$\nu_{\mathcal{S}}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is visible from the camera,} \\ 0 & \text{otherwise.} \end{cases} \qquad (4)$$

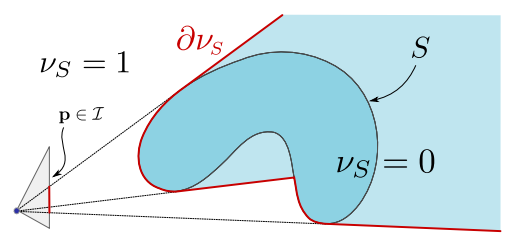


Figure 1: *Banana Shape* seen from a vantage point (See [14]). The energy defined over the image explains the visibility interface $\partial\nu_{\mathcal{S}}$ (in red) of the surface $\mathcal{S}$.

Finally, by using the separation technique proposed by [47], we can rewrite energy (3) as an integral over only the visible surface:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\,. \qquad (5)$$

To obtain $\mathbf{g}$ from $\overline{\mathbf{g}}$, we refer the reader to Section 5.3.2 and especially Equation (49), as well as [47], which comprehensively details this step. Figure 1 illustrates the case of an energy defined over a visible volume.

Let us emphasize that handling properly the visibility term is a non-trivial undertaking. In particular, this is one of the major difficulties in the stereovision problem. Previous works cope with this difficulty more or less elegantly. Most often the authors approximate the visibility in some pre-processing steps which can be completely prior to the whole algorithm or else inside the iterations of the minimization process [3, 13, 20, 22, 34, 39, 50].

Only recently, some authors [15, 47] manage to rigorously and fully account for visibility in the optimization

process. In their recent work, Yezzi and Soatto [47] (for convex surfaces) and Gargallo et al. [15] (in the general case) provide the exact gradient of the reprojection error. This computation is done in the mathematical framework of continuous (smooth) surfaces as used in [42]. This work has shown that a proper handling of the visibility automatically forces the apparent contours generated by the reconstructed surface to perfectly match with the apparent contours in the observed data. This makes the use of additional energy terms like ballooning [44], visual hull [19, 37] or contours unnecessary and significantly reduces the minimal surface bias present in many other problem formulations.

In this paper, we rigorously minimize energies (1) and (5). But contrary to the works [15, 47] which deal with continuous surfaces, here we consider triangle mesh-based surfaces.

1.2 Gradient Descent Optimization

Energy (5) is rather difficult to optimize. In particular, the complexity is due to the dependency on the normal and the visibility term. There exist several tools to minimize energy functionals. Recent methods, such as graph-cuts for example, allow us to find global minimum. At the present time these global optimization techniques are limited to rather simple energies [28] and, based on the current state-of-the-art, it seems extremely difficult to apply them for minimizing energies such as (5). Recent advances also allow such minimization via total variation and convex relaxation methods [24, 26], however it is difficult to apply those methods for functionals depending on the surface normal. The recent work of [27] allows to take the normal into account but is not directly applicable to general energy functionals like (1) and (5). Taking the normal into account in the minimization allows to produce high quality 3D models [31]. In this paper we compute the derivatives of the generic functionals (1) and (5) which allow us to minimize it via gradient descents [6, 13, 16, 42], see Section 2.

1.3 Triangle Mesh-based Representation

Surface representation based on polyhedral and especially triangular meshes are the most commonly used in graphics. Moreover, the design of graphics hardware makes triangular meshes the most natural way to represent surfaces in a number of applications. In the variational framework, this type of discrete representations leads to the Lagrangian setting. Lagrangian methods are generally contrasted with Eulerian approaches which are mainly based on the level set representation [32]. During the last decades, Eulerian methods have become very popular mainly because they allow us to naturally deal with topological changes. In particular, this last setup has been extensively used for 3D reconstruction problems [5, 13, 15, 22, 23, 25, 29, 35, 40, 41, 46, 49]. Nevertheless, recent advances in mesh processing allow Lagrangian methods to enjoy the same facilities [33, 51]. In other respects, in the Eulerian methods, the gradient is computed in the continuous framework. Technically, computing the gradient in the continuous framework is more complicated than in a discrete framework, since the first one requires functional analysis when the second one only needs differential calculus. Furthermore, in practice in Eulerian approaches, one finally needs to discretize the continuous gradient flow since the level set function is also discretized on a grid. Also, this discretization (which is usually obtained using discrete differential operators [30] or finite element modeling [11]) is sometimes difficult to obtain, as we can see for example in Sections 3 and 4. By directly considering discrete surfaces, this last step is not necessary in our case.

Now, let us note that Langragian methods may be classified into two different approaches: The first strategy consists in 1) formulating the problem with continuous surfaces and computing the gradient in the continuous framework, then 2) discretizing the continuous gradient in order to apply it to the discrete surface. The second strategy consists in 1) formulating directly the problem with the discrete surface representation, then 2) computing the exact gradient of the formulated energy. Clearly, the second strategy is better than the first one: in fact in the first strategy we do not know if finally in practice the discrete representation minimizes something; in the best case, we do not know which exact energy the computed solution really minimizes since it does not necessarily corresponds to the representation. Also, surprisingly, a number of works follow the first strategy, see for example [3, 10, 51]. In particular, the gradient is computed using normal velocities whereas the second approach may have tangential components which leads to more coherent

flow. In this paper, we follow the second strategy as other authors: for example, Slabaugh and Unal [38] who deal with surface segmentation, Eckstein et al. [12] who are interested in shape matching, or Vu et al. [45] who proposes a complete multi-view stereovision algorithm. We also use the similar strategy presented by Debreuve et al. [7] who deal with discrete parametric active contours for segmentation or Dziuk and Elliott [11] using finite element modeling on evolving surfaces. More exactly, we detail the exact gradient flow of energy (1) in which the surface $\mathcal{S}$ is explicitly a discrete surface based on triangular meshes, and also extend it to visibility-driven energies (5).

1.4 Contributions

In this article, we first give an overview of gradient descent flows with deforming surfaces, when represented by triangular surface meshes. Here, the gradient is the one of the energy defined with the discrete surface; we do not need to approximate and discretize it when we finally evolve the surface. Even though the visibility plays a key role in computer vision, until now it has been managed more or less elegantly. It is clearly a key difficulty in the field, which has been recently solved in the theory of continuous surfaces [14, 15]. Here, we show how to rigorously deal with the visibility in the framework of discrete surface representations and we give the gradient flow of generic energies which encompasses a large class of energies used in computer vision (see Section 4).

We then illustrate the presented results by giving the gradient of commonly used functionals in computer vision and graphics, and we emphasize the 3D reconstruction applications for which the visibility is fundamental; we thus show in Section 5 how one can apply mesh evolution techniques to 3D reconstruction applications from multiple views. In particular in Section 5.3 we focus on the multi-view stereovision problem and we propose a successful algorithm which, since it fully accounts for visibility, automatically aligns contour generators with image contours.

This article generalizes our previous conference paper [8] in which the considered energy does not depend on the normal. This allows us to present here a larger spectrum of applications including multi-view shape from shading and multi-view photometric stereo (see Section 5.5).

2 Gradient Flows

In this section, we are interested in minimizing energy functionals defined on surfaces with respect to the surface representation. Whatever surface representation one chooses, the energy minimization should be consistent with it in order to be sure the energy minimized is the right one. It has been common to minimize such energy by performing gradient descent. Computing an adequate gradient corresponding to the representation of the surface is not trivial, and the following gives a way for computing generalized gradient flows for an energy $E(\mathcal{S})$, where $\mathcal{S}$ is a 2-dimensional surface in $\mathbb{R}^3$, g is a differentiable scalar function and $\mathbf{n}$ is the Gauss map for $\mathcal{S}$.

In the following, we explain how to obtain such gradients firstly in the theory of continuous smooth surfaces, and finally using triangular meshes.

2.1 Gradient Descent in the Continuous Case

Let M denote the set of all admissible 2D-manifolds embedded in $\mathbb{R}^3$, and $\mathcal{S} \in M$. Let $\mathbf{v}$ be a vector in the tangent space of M, denoted by $T_F M$, associated with an inner product $\langle \cdot, \cdot \rangle_F$. Let $E(\mathcal{S}) : M \to \mathbb{R}$ be a surface functional as defined previously (1) such that its Gâteaux Derivative in the direction $\mathbf{v}$ can be expressed as : $DE(\mathcal{S}, \mathbf{v}) \equiv \frac{\mathrm{d}}{\mathrm{d}\tau} E(S + \tau \mathbf{v})\Big|_{\tau=0}$. Then the gradient of E at $\mathcal{S}$ is the unique vector $\nabla_M E(\mathcal{S}) \in T_F M$ such that $DE(\mathcal{S}, \mathbf{v}) = \langle \nabla_M E(\mathcal{S}), \mathbf{v} \rangle_F$ for all $\mathbf{v} \in T_F M$. See for instance Solem and Overgaard [42] for a more detailed explanation.

Then, the gradient descent flow of an energy $E(\mathcal{S})$ as the form:

$$\begin{cases} \mathcal{S}(0) = \mathcal{S}^0 \,, \\ \dfrac{\partial \mathcal{S}(t)}{\partial t} = -\nabla_M E(\mathcal{S}(t)) \,. \end{cases} \tag{6}$$

2.2 Gradient Descent for Polyhedral Meshes

In practice, we often deal with discrete representations of the surface. Also whichever this representation is, computing exactly the gradient of the energy including directly the discrete representation of the surface is much

more suitable than computing the gradient in an ideal continuous framework (with continuous surfaces) and then discretizing the continuous gradient accordingly to the discrete representation. In fact, the second strategy has two significant drawbacks. First, the continuous gradient lets often appear terms that are very difficult to discretize and which sometimes do even not really make sense in the case of discrete surfaces. For example the notion of surface curvature on mesh representation has a lot of different approximations. Second, since the discrete object we are practically handling is not deformed following the exact gradient but by an approximation of it, finally, we do not exactly know what we are minimizing. In this paper, we then compute the exact gradient of the energy including directly the discrete representation of the surface in the same way as [11, 12].

Let the mesh $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ be the piecewise planar polyhedral representation of $\mathcal{S}$. Vertices of $\mathbf{X}$ are denoted by $\mathbf{x}_k$ and $\mathcal{S}$ is deformed by moving vertices $\mathbf{x}_k$. We denote by $\phi_k : \mathcal{S} \to \mathbb{R}$ the piecewise linear, interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i \neq k$. Then any point $\mathbf{x}$ on the surface $\mathcal{S}$ can be defined such that $\forall \mathbf{x} \in \mathcal{S}$, $\mathbf{x} = \sum_k \mathbf{x}_k \phi_k(\mathbf{x})$, where we also have $\forall \mathbf{x} \in \mathcal{S}$, $\sum_k \phi_k(\mathbf{x}) = 1$.

Let the set $\{\mathbf{V}_k\}$ be a parametrized vector field defined on all the vertices $\mathbf{x}_k$ of the mesh $\mathbf{X}$ representing the surface deformation. $\{\mathbf{V}_k\}$ can be naturally extended on $\mathcal{S}$ by a piecewise linear vector field on $\mathcal{S}$. For convenience, we denote this extension $\mathbf{V} : \mathbf{V}(\mathbf{x}) = \sum_k \mathbf{V}_k \phi_k(\mathbf{x})$.

Then, evolving $\mathbf{X}$, by moving its vertices $\mathbf{x}_k$ following $\mathbf{V}_k$ is equivalent to deform the surface $\mathcal{S}$ by the dense deformation following $\mathbf{V}$.

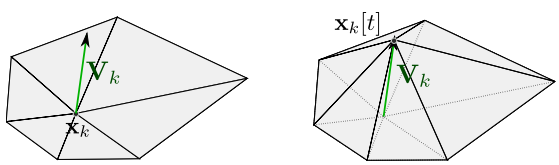


Figure 2: Local evolution of a surface point $\mathbf{x}_k$ under induced velocity $\mathbf{V}_k$ used to compute the gradient.

The gradient of the energy is computed using shape gradient [7]. We consider the evolution of this energy according to the deformation $\mathbf{V}$. In other words, we assume that the vertices $\mathbf{x}_k[t]$ of $\mathbf{X}[t]$ are moving according to $\mathbf{x}_k[t] = \mathbf{x}_k^0 + t\mathbf{V}_k$ (See Figure 2). The method for computing the gradient of $E(\mathcal{S})$ consists in computing the directional derivative of $E(\mathcal{S}[t])$ for this deformation and then in rewriting it as a scalar product of $\mathbf{V}$, i.e. as $\langle \mathbf{V}, \mathbf{G} \rangle$, $\mathbf{G}$ being independant of $\mathbf{V}$. The obtained vector $\mathbf{G}$ is called the *gradient* and the energy necessarily decreases when deforming the surface according to its opposite direction $-\mathbf{G}$. Indeed, for $\mathbf{x}_k[t] = \mathbf{x}_k^0 - t\mathbf{G}_k$, we have

$$[E \circ \mathcal{S}]'(0) = -\langle \mathbf{G}, \mathbf{G} \rangle \leq 0,$$

see [12].

Let us recall now that, as underlined by [6, 12], the notion of gradient depends on the underlying scalar product. In this work we will only consider the L^2 inner product which has the advantage of taking into account the area of the triangles contrarily to the pointwise scalar product, which is necessary if the surface is not a regular mesh. Let $A = \{\mathbf{a}_k\}$ and $B = \{\mathbf{b}_k\}$ be vector fields on the mesh $\mathbf{X}$. Let a and b be their linear extension on the whole surface $\mathcal{S}$. Then their L^2 scalar product is:

$$\begin{aligned} \langle A, B \rangle_{L^2} &= \int_{\mathcal{S}} \langle a(\mathbf{x}), b(\mathbf{x}) \rangle \, \mathrm{d}s \\ &= \int_{\mathcal{S}} \left\langle \sum_k \mathbf{a}_k \phi_k(\mathbf{x}), \sum_k \mathbf{b}_k \phi_k(\mathbf{x}) \right\rangle \mathrm{d}s \\ &= A^T \, M \, B \, , \end{aligned} \tag{7}$$

where $M = \{m_{ij}\}$ is the mass matrix defined by $m_{ij} = Id_3 \int_{\mathcal{S}} \phi_i(x)\phi_j(x)\mathrm{d}s$. In the last line of Equation (7), A and B are the matricial representations of the vector fields. They are column vectors containing successively $\mathbf{a}_k$ and $\mathbf{b}_k$ vectors. Then the gradient becomes:

$$\nabla E(\mathbf{X}) = M^{-1} \frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}) \, , \tag{8}$$

where

$$\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}) = \left[\frac{\partial E}{\partial \mathbf{x}_1}(\mathbf{X}) \, \frac{\partial E}{\partial \mathbf{x}_2}(\mathbf{X}) \dots \frac{\partial E}{\partial \mathbf{x}_n}(\mathbf{X}) \right]'$$

corresponds to the gradient associated with the pointwise inner product $< A, B >= \sum_k \mathbf{a}_k \cdot \mathbf{b}_k$.

One classically approximates M by the diagonal mass lumping $\tilde{M}$, where $\tilde{m}_{ii}$ is the area of the Voronoi dual

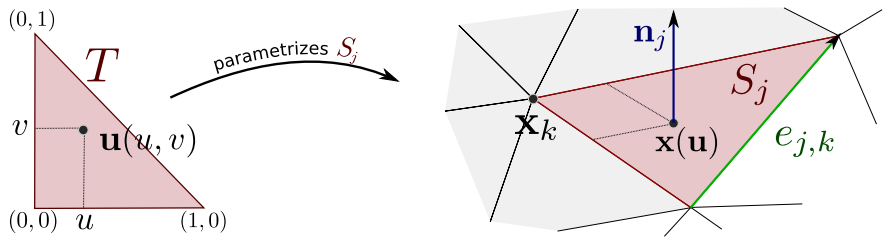


Figure 3: Local parametrization of the discrete representation of the surface into a triangle mesh. Figure on the left shows the local parametrization $\mathbf{u}(u, v)$ of a surface point. On the right, we show the notations used in the paper where $\mathbf{x}_k$ is the current vertex of $\mathcal{S}$, $\mathcal{S}_j$ is the current facet around $\mathbf{x}_k$ and $\mathbf{e}_{j,k}$ is the opposite edge of $\mathbf{x}_k$ in $\mathcal{S}_j$. The gray area represent the set of triangles around $\mathbf{x}_k$ denoted by $\mathcal{J}_k$.

cell of $\mathbf{x}_i$ times the identity matrix Id_3, see e.g. [12]. It follows that the L^2 gradient descent flow is:

$$\begin{cases} \mathbf{X}[0] = \mathbf{X}^0 \,, \\ \dfrac{\partial \mathbf{X}[t]}{\partial t} = -\tilde{M}^{-1} \dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X}[t]) \,. \end{cases} \tag{9}$$

2.2.1 Triangle Mesh Representation and Notations

The previous results are valid for any polyhedral representation. Also in practice and in the following, we will focus on triangle representation that are easier to understand and more simple to handle using computers.

Let $\mathcal{S}_j$ be the jth triangle of the mesh and $\mathbf{x}_k$ be a vertex of $\mathcal{S}_j$. Let us consider the parametrization on the triangle $\mathcal{S}_j$ such that

$$\mathbf{x}(\mathbf{u}) = \mathbf{x}_k + u\, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} + v\, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}} \,, \tag{10}$$

where $\mathbf{x}_{k1}$ and $\mathbf{x}_{k2}$ are the two other vertices of the triangle $\mathcal{S}_j$ such that $(\mathbf{x}_k, \mathbf{x}_{k1}, \mathbf{x}_{k2})$ is a counter-clockwise triangle. Here, $\mathbf{u} = (u, v) \in T$, where

$$T = \{(u, v) | u \in [0, 1] \text{ and } v \in [0, 1 - u]\} \,.$$

Figure 3 illustrates this representation and parametrization. Too be rigorous, we should write $\mathbf{x}_k^j(\mathbf{u})$ instead of $\mathbf{x}(\mathbf{u})$, since the parametrization depends on j and k. Nevertheless, in order to simplify equations and improve the clarity of the paper, we remove these indexes in the rest of the paper. In the following, when we use $\mathbf{x}(\mathbf{u})$, the choice of the associated j and k is directly given by the context. On each triangle $\mathcal{S}_j$, we denote by ϕ_k the linear interpolating basis function that verifies $\phi_k(\mathbf{x}(\mathbf{u})) = (1 - u - v)$, $\forall (u, v) \in T$.

We denote by A_j the surface area of triangle $\mathcal{S}_j$ and by $\mathbf{n}_j$ its outward surface normal. A_j and $\mathbf{n}_j$ can easily be defined with respect to the triangle vertices such that:

$$A_j = \frac{1}{2} \left| \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}} \right| \quad \text{and} \quad \mathbf{n}_j = \frac{\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}}{2\, A_j} \,,$$

where the operator $\wedge$ denotes the cross product. Indeed it is easy to show that the area surface measure on the surface $\mathrm{d}s$ can be written using the parametrization $\mathbf{u}$ such that:

$$\mathrm{d}s = 2\, A_j\, \mathrm{d}\mathbf{u} \,. \tag{11}$$

In the following, we also denote by $\mathbf{e}_{j,k}$ the opposite edge of the vertex $\mathbf{x}_k$ in the triangle $\mathcal{S}_j$ such that $\mathbf{e}_{j,k} = \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}$.

$\mathcal{J}_k$ represents the set of triangles containing vertex $\mathbf{x}_k$ and the set $\mathcal{K}_j$ is the set of indexes of the three vertices of triangle $\mathcal{S}_j$ (See Figure 3.).

3 Gradient of Weighted Area Functionals

3.1 Continuous Case

Let $\mathcal{S} \in M$ be the surface to deform in order to minimize the following classical weighted area functional (1):

$$E(\mathcal{S}) = \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x})) \mathrm{d}s \,,$$

where g is a differentiable scalar function defined all over the surface. Then using shape gradient as described previously, one can rewrite the differential of $E(\mathcal{S})$ under a linear deformation V in order to find the expression of $\nabla E(\mathcal{S})$. As used in [15] and shown in [42], the gradient descent flow of the functional defined in (1) has the form:

$$\nabla E(\mathcal{S}) = \nabla \cdot (g_{\mathbf{n}} + g\mathbf{n}) \ , \tag{12}$$

where $g_{\mathbf{n}}$ is the gradient on the unit sphere $\mathbb{S}$.

3.2 Gradient for Triangle Mesh-based Surfaces

In this paragraph, we consider the discretization on the surface and the gradient descent flow described in the previous section, when the surface is represented by a triangular mesh. Also we consider the case where the energy functional to be minimized is:

$$\begin{aligned} E(\mathcal{S}) &= \int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, \mathrm{d}s \\ &= \sum_j 2\, A_j \int_T g(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\, \mathrm{d}\mathbf{u} \ , \end{aligned} \tag{13}$$

where A_j and $\mathbf{n}_j$ are defined in Section 2.2.1.

Let us focus on the evolution of $E(S)$ under the induced velocity $\mathbf{V}$ on triangle $\mathcal{S}_j$ only. We have

$$E(\mathcal{S}_j[t]) = 2\, A_j[t] \int_T g(\mathbf{x}(\mathbf{u}) + t\mathbf{V}(\mathbf{x}(\mathbf{u})), \mathbf{n}_j[t])\, \mathrm{d}\mathbf{u} \ , \tag{14}$$

where $A_j[t]$ is the area of the triangle $\mathcal{S}_j[t]$ and $\mathbf{n}_j[t]$ is its normal at time t. By simple derivation we get

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0} &= 2\, A_j'[0] \int_T g(\mathbf{x}, \mathbf{n}_j)\, \mathrm{d}\mathbf{u} \\ &+ 2\, A_j \int_T \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) \cdot \mathbf{V}(\mathbf{x})\, \mathrm{d}\mathbf{u} \\ &+ 2\, A_j \int_T \nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j) \cdot \mathbf{n}_j'[0]\, \mathrm{d}\mathbf{u} \ . \end{aligned} \tag{15}$$

Above, in order to simplify equations, we have removed the dependency in $\mathbf{u}$ by writing $\mathbf{x}$ instead of $\mathbf{x}(\mathbf{u})$. In the sequel, we will use this abuse of notation. In order to rewrite $\frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0}$ as a scalar production of $\mathbf{V}$, we use also the fact that $\forall \mathbf{x} \in \mathcal{S}_j$, $\mathbf{V}(\mathbf{x}) = \sum_{k \in K_j} \mathbf{V}_k \phi_k(\mathbf{x})$, and then use the expressions of $A_j'[0]$ and $\mathbf{n}_j'[0]$ computed in appendices A.1 and A.2:

$$\begin{aligned} A_j'[0] &= \frac{\mathrm{d}}{\mathrm{d}t} A_j[t]\Big|_{t=0} = \frac{1}{2}\, \left(\mathbf{n}_j \wedge \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}\right) \cdot \mathbf{V}_k \ , \\ \mathbf{n}_j'[0] &= \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{n}_j[t]\Big|_{t=0} = \frac{\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k - \left(\left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k\right) \cdot \mathbf{n}_j\right) \mathbf{n}_j}{2\, A_j} \ . \end{aligned} \tag{16}$$

It follows that:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}_j[t])\Big|_{t=0} = \sum_{k \in K_j} \mathbf{V}_k \cdot \Bigg\{ & \frac{\mathbf{n}_j \wedge \mathbf{e}_{j,k}}{2A_j} \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\, \mathrm{d}s \\ & + \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) \phi_k(\mathbf{x})\, \mathrm{d}s \\ & - \frac{\mathbf{e}_{j,k}}{2\, A_j} \wedge \int_{\mathcal{S}_j} g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j)\, \mathrm{d}s \Bigg\} , \end{aligned} \tag{17}$$

where we define $g_{\mathbf{n}} = \nabla_n g(\mathbf{x}, \mathbf{n}_j) - \langle \nabla_n g(\mathbf{x}, \mathbf{n}_j), \mathbf{n}_j \rangle\, \mathbf{n}_j$, where $\nabla_n g(\mathbf{x}, \mathbf{n}_j)$ is the gradient of g with respect to the second variable (i.e. $\mathbf{n} \in \mathbb{R}^3$).

It then immediately follows that

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} E(\mathcal{S}[t])\Big|_{t=0} = \sum_k \mathbf{V}_k \cdot \Bigg[\sum_{j \in \mathcal{J}_k} \Bigg\{ & \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) \phi_k(\mathbf{x})\, \mathrm{d}s \\ & - \frac{\mathbf{e}_{j,k}}{2A_j} \wedge \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\, \mathbf{n}_j + g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j)\, \mathrm{d}s \Bigg\} \Bigg] , \end{aligned} \tag{18}$$

Finally the part in brackets gives the kth component of $\frac{\partial E}{\partial \mathbf{X}}$ and one has to use Equation (9) in order to get the gradient and optimize the mesh $\mathbf{X}$.

3.3 Comparison of the Continuous Gradient with the Gradient for Triangle Meshes

By looking at both gradients, one may note similarities. First it is worth it to notice that the discrete gradient is written with respect to well defined quantities that can be easily expressed, such as edges vectors, or triangle area.

On the other hand the continuous gradient has terms depending on the curvature for instance, which is well defined in the theory of continuous differential surfaces, but is hard to discretize on mesh representations. By making a piecewise linear surface assumption, the obtained gradient directly accounts for those intrinsic properties.

4 Gradient of Functionals defined on Visible Surface

As described in the introduction, computer vision applications are most likely able to deal with what the camera sees, i.e. the projected image. Such projection imply $3D$ information such as depth and occlusion, and a camera model. An energy functional can be expressed accordingly as in Equation (5), where the energy can be defined on the visible interface (See also Figure 1). Often some quantities can be estimated (it can be for instance color, photometric normals, reflectance, etc) and used to compare them with data in the input images. In the sequel we denote such energies reprojection error functionals. As shown for example in [15, 48], an accurate Bayesian formulation of computer vision problems (as e.g. in 3D reconstruction) yields the minimization of some reprojection error functionals instead of classical weighted area functionals.

Let us then consider the energy functional (5)

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\,.$$

In this section, we compute the gradient of this functional with respect to the shape $\mathcal{S}$.

In practice, the direction of the 3D vector $\mathbf{g}$ corresponds to the direction of the viewing ray. Thus the reprojection error functionals generally verify $\mathbf{g}(\mathbf{x}, \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})) \cdot \mathbf{n}_{\partial\mathcal{V}}(x) = 0$ for all points $\mathbf{x}$ on the horizon of the visibility interface ($\partial\mathcal{V}$ being the visibility interface and $\mathbf{n}_{\partial\mathcal{V}}$ its normal, see Figure 1). Then, by Gauss' divergence theorem, we can rewrite $E(\mathcal{S})$ as an integral over $\mathbb{R}^3$, see [14, 15].

$$\begin{aligned} E(\mathcal{S}) &= \int_{\partial\mathcal{V}} \mathbf{g}(\mathbf{x}, \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})) \cdot \mathbf{n}_{\partial\mathcal{V}}(\mathbf{x})\, \mathrm{d}s \\ &= -\int_{\mathbb{R}^3} \nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x}\,. \end{aligned} \tag{19}$$

Let $\mathcal{S}[t]$ be a variation of $\mathcal{S}$ such that $\mathcal{S}[t] = \mathcal{S} + t\,V$. By the product rule, the derivative of the energy with respect to t is

$$\begin{aligned} \frac{d}{dt} E(\mathcal{S}[t])\Big|_{t=0} = &-\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\Big(\mathbf{g}(\mathbf{x}, \mathbf{n}[t](\mathbf{x}))\Big)\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x} \\ &- \int_{\mathbb{R}^3} \nabla \cdot \Big(\mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))\Big)\, \frac{d}{dt} \nu_{\mathcal{S}[t]}(\mathbf{x})\Big|_{t=0} \mathrm{d}\mathbf{x}\,. \end{aligned} \tag{20}$$

For notation simplicity we have denoted $\mathcal{S} = \mathcal{S}[0]$, $\mathbf{n} = \mathbf{n}[0]$.

In Section 4.2.1, we write the first integral of Equation (20) linearly with respect to $\mathbf{V}$. We then describe the corresponding part of the gradient of $E(\mathcal{S})$. We denote it $\mathbf{G}^{norm}$.

In the second integral of Equation (20), the normal does not depend on t. Thus, this term is the derivative of a quantity (which does not depend on $\mathcal{S}$ and does not vary in time) integrated over the visible volume. In other words, the second term of Equation (20) is the derivative (with respect to the shape $\mathcal{S}$) of an energy

$$\tilde{E}(\mathcal{S}) = \int_{\mathbb{R}^3} f(\mathbf{x}) \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}\mathbf{x}\,,$$

where $f(\mathbf{x})$ does not depend on $\mathcal{S}$;

$$f(\mathbf{x}) = -\nabla \cdot \Big(\mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))\Big).$$

It is the derivative of a scalar field integrated over a visible volume.

Until now all equations are valid for both the continuous case and the discrete case. In order to find the gradient expression, the idea is to see what happens to the energy under an induced velocity on the surface. In the continuous case under a continuous vectorial field V, and under a piecewise linear vectorial field $\mathbf{V}$ defined by the discrete field $\{V_k\}$ in the other case. In both cases we have the following geometric interpretation of the changes in the energy when the surface is deforming.

Since this energy $\tilde{E}(\mathcal{S})$ is an integral over the visible volume, its variations are only due to the variations of the visible volume. Also, as illustrated by Figure 4 (reduced to the 2D case for simplicity), when the vertex $\mathbf{x}_k$ is moving according to $\mathbf{V}_k$ we have to separate two cases:

1. when all the triangles adjacent to $\mathbf{x}_k$ are visible, the variation of the visible volume is just the sum of the

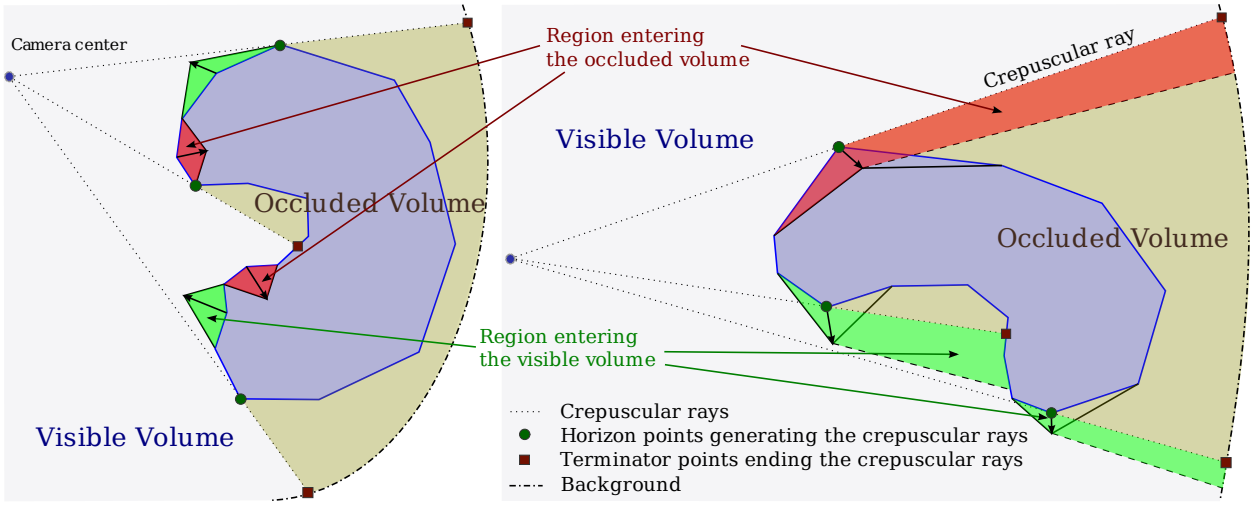


Figure 4: Geometric representation of the change of visibility when moving the mesh. Contrary to the interior (left), movements of the horizon (right) strongly affect the movement of the visible interface between visible and occluded volumes by creating a movement of the crepuscular rays.

tetrahedron formed by the adjacent triangles and the moved point $\mathbf{x}_k + \mathbf{V}_k$ (see Figure 4, left). The corresponding gradient computation is detailed in Section 4.2.2. By replacing $f(\mathbf{x})$ by $-\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$, this gives a second part of the gradient of $E(\mathcal{S})$; we denote it $\mathbf{G}^{int}$.

2. when $\mathbf{x}_k$ is generating occluding contours in images, i.e. when it is a *horizon* point, its movement affects the visibility of other points located behind it (called *terminator* points). So the variation of the visible volume is the sum of the first case term, plus the volume swept out by the crepuscular rays generated by the horizon movement (see Figure 4, right). The corresponding gradient computation is detailed in Section 4.2.3. Again, by replacing $f(\mathbf{x})$ by $-\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$, this gives a third part of the gradient of $E(\mathcal{S})$ that we denote it $\mathbf{G}^{horiz}$.

Now let us first remind the expression of the gradient in the continuous case (Section 4.1) and then compute it when using triangle meshes (Section 4.2).

4.1 Gradient in the Continuous Case

The differential of the energy (5) in the case of continuous surfaces is the work of Gargallo [14] and is:

$$\nabla \cdot \Big(\mathbf{g_n} \cdot \mathbf{n} + \mathbf{g}\Big)\nu_{\mathcal{S}} - \mathbf{x}^t \nabla \mathbf{n}\,(\mathbf{g} - \mathbf{g}')\delta(\mathbf{x} \cdot \mathbf{n})\nu_{\mathcal{S}}\ , \quad (21)$$

where δ is a Dirac distribution function and $\mathbf{g}'$ is the value of $\mathbf{g}$ at the terminator of the current point. In the presence of a discrete surface, terms like the curvature $\nabla \mathbf{n}$ are difficult to handle and have to be approximated. The following section shows that by computing the gradient with respect to the true representation of the surface, one can use a new formulation using intrinsic surface properties.

4.2 Gradient for Triangle Meshes

Here, we derive the exact gradient (with respect to the shape) of these functionals in the case where the surface is represented by a triangular mesh. Here, contrary to our previous conference article [8], the function $\mathbf{g}$ may also depends on the normal of the surface.

Then we can summarize the way gradient descent is performed in that case. The variation of an energy that depends on the visibility and the surface normal can be

decomposed into three different cases. The three terms are:

1. the one due to the change in the normal $\mathbf{G}^{norm}$ (Equation (25)) that corresponds to Section 4.2.1,

2. the term due to the movement of points on the fully visible areas $\mathbf{G}^{int}$ (Equation (30)) of Section 4.2.2,

3. and finally the term due to the movement of points on occluding contours $\mathbf{G}^{horiz}$ (Equation (36)) that makes global (in the sense on the whole surface) changes of the energy (Described in Section 4.2.3).

4.2.1 Term due to the Variation of the Normal

In this section, we are going to rewrite the first integral term appearing in equation (20), linearly with respect to $\mathbf{V}$. This will directly give us the first term of the gradient of $E(\mathcal{S})$ defined in (5). First, using Gauss' divergence theorem again, let us rewrite it as an integral over the surface. This gives:

$$\begin{aligned}
&-\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\mathbf{g}(\mathbf{x},\mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \int_{\mathcal{S}} \frac{d}{dt}\mathbf{g}(\mathbf{x},\mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \cdot \mathbf{n}(\mathbf{x})\,\nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s \\
&= \sum_j \int_{\mathcal{S}_j} \frac{d}{dt}\mathbf{g}(\mathbf{x},\mathbf{n}_j[t])\Big|_{t=0} \cdot \mathbf{n}_j\,\nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s \\
&= \sum_j \int_{\mathcal{S}_j} (D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)\mathbf{n}_j'[0]) \cdot \mathbf{n}_j\,\nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s\,,
\end{aligned} \tag{22}$$

where $D_{\mathbf{n}}$ is the differential with respect to the second variable and where $\mathbf{n}_j'[0]$ is the derivative of the normal $\mathbf{n}_j[t]$ of the triangle $\mathcal{S}_j[t]$ at time $t=0$. Here we have assumed that the reprojection error functional verifies

$$\left(\frac{d}{dt}\mathbf{g}(\mathbf{x},\mathbf{n}_{\partial\mathcal{V}}[t](\mathbf{x}))\right) \cdot \mathbf{n}_{\partial\mathcal{V}}(x) = 0$$

for all points $\mathbf{x}$ on the horizon lines of the visibility interface; it is generally the case in practice, see [14, 15].

The computation of $\mathbf{n}_j'[0]$ is detailed in appendix A.2. We have

$$\mathbf{n}_j'[0] = \frac{1}{2A_j}\left(\left(\sum_{k\in K_j}\mathbf{e}_{j,k}\wedge\mathbf{V}_k\right) - \left((\sum_{k\in K_j}\mathbf{e}_{j,k}\wedge\mathbf{V}_k)\cdot\mathbf{n}_j\right)\mathbf{n}_j\right). \tag{23}$$

So

$$\begin{aligned}
&\int_{\mathbb{R}^3} \nabla \cdot \frac{d}{dt}\mathbf{g}(\mathbf{x},\mathbf{n}[t](\mathbf{x}))\Big|_{t=0} \nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \sum_j \int_{\mathcal{S}_j} \mathbf{n}_j'[0]\cdot D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)^T\mathbf{n}_j\,\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s \\
&= \sum_k \mathbf{V}_k \cdot \Big[\sum_{j\in J_k}(-1)\frac{\mathbf{e}_{j,k}}{2\,A_j}\wedge \\
&\quad \int_{\mathcal{S}_j} P_{n_j\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)^T\mathbf{n}_j)\,\nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s\Big]\,,
\end{aligned} \tag{24}$$

where $P_{n_j\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)^T\mathbf{n}_j)$ is the projection on the orthogonal plane to $\mathbf{n}_j$ of $D_{\mathbf{n}}\mathbf{g}(\mathbf{x},\mathbf{n}_j)^T\mathbf{n}_j$. $\mathbf{e}_{j,k}$ is defined in appendix A.1. Roughly $\mathbf{e}_{j,k}$ is the opposite edge of the vertex $\mathbf{x}_k$ in the triangle $\mathcal{S}_j$. Finally, writing the previous expression on the mesh parametrization $\mathbf{x}(\mathbf{u})$ yields: which can be rewritten as :

$$\begin{aligned}
\mathbf{G}_k^{norm} = &\sum_{j\in J_k}\mathbf{e}_{j,k}\wedge \\
&\int_T P_{n_j\top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}(\mathbf{u}),\mathbf{n}_j)^T\mathbf{n}_j)\,\nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))\,\mathrm{d}\mathbf{u}.
\end{aligned} \tag{25}$$

4.2.2 Term Due to the Tetrahedra of the Visible Adjacent Triangles

In this paragraph, let us focus on the variation of the energy caused by the variation of the visible volume corresponding to the tetrahedra formed by the visible adjacent triangles of vertex $\mathbf{x}_k$ and $\mathbf{x}_k+\mathbf{V}_k$. (See Figure 2)

In the sequel, we denote $\mathcal{S}_j$ the jth triangle of the mesh. Following the variation of the energy caused by the visible adjacent triangles of $\mathbf{x}_k$, we have

$$\tilde{E}(\mathcal{S}[t])-\tilde{E}(\mathcal{S}[0]) = \sum_j \varpi_{j,\mathbf{V}_k}\int_{Vol[j,\mathbf{V},t]} f(\mathbf{x})\,\mathrm{d}\mathbf{x}+\dots\,, \tag{26}$$

where $Vol[j,\mathbf{V},t]$ is the volume of the tetrahedron formed by the vertices of the *visible* triangle $\mathcal{S}_j$ and the

point $\mathbf{x}_k + \mathbf{V}_k$. (See Figure 2) The sign $\varpi_{j,\mathbf{V}_k}$ specifies if matter has been added to or removed from the object volume and it is equal to the sign of $\mathbf{n}_j \cdot \mathbf{V}_k$, where $\mathbf{n}_j$ is the outward surface normal of triangle $\mathcal{S}_j$. The dots part "..." on the right of equation (26) is null except if the vertex $\mathbf{x}_k$ is a horizon point, which means that this point is on the occluding contour; this additional part will be detailed in the next paragraph.

Now, we parametrize the volume $Vol[j, \mathbf{V}, t]$ by the point $\mathbf{x}(u, v, w) = \mathbf{x}(u, v) + w\ t\mathbf{V}_k$, where $\mathbf{x}(u, v) = \mathbf{x}(\mathbf{u})$ parametrizes the triangle $\mathcal{S}_j$ as defined previously and shown in figure 3; The local parametrization is such that $\mathbf{u}(u, v) \in T = \{(u, v) \mid u \in [0, 1]$ and $v \in [0, 1 - u]\}$ and $w \in [0, 1 - u - v]$. By a change of variables, Equation (26) becomes

$$\sum_j \varpi_{j,\mathbf{V}_k} \int_T \int_0^{\phi_k(\mathbf{u})} \Big\{ f(\mathbf{x}(\mathbf{u}) + wt\mathbf{V}_k) \times |det(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}}, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}, t\mathbf{V}_k)| \Big\} \, \mathrm{d}w\mathrm{d}\mathbf{u} + \ldots \quad (27)$$

Let A_j be the area of the triangle $\mathcal{S}_j$. It is easy to show that $\varpi_{j,\mathbf{V}_k} |det(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}}, \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}, t\mathbf{V}_k)| = 2t\, A_j \mathbf{V}_k \cdot \mathbf{n}_j$. Then Equation (27) becomes

$$2t\, \mathbf{V}_k \cdot \left[\sum_j A_j \mathbf{n}_j \int_T \int_0^{\phi_k(\mathbf{u})} f(\mathbf{x}(\mathbf{u}) + wt\mathbf{V}_k)\, \mathrm{d}w\mathrm{d}\mathbf{u} \right] + \ldots \quad (28)$$

It follows that the limit of $\frac{\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0])}{t}$ when t tends to zero is

$$\mathbf{V}_k \cdot \left[2 \sum_j A_j \mathbf{n}_j \int_T f(\mathbf{x}(\mathbf{u}))\, \phi_k(\mathbf{x}(\mathbf{u}))\, \mathrm{d}\mathbf{u} \right]. \quad (29)$$

Now the derivative of the energy can be expressed as a scalar product between the velocity $\mathbf{V}_k$ and a quantity that corresponds to the gradient as explained previously. The part in square brackets corresponds then to the interior term of the gradient of $E(\mathcal{S})$ with respect to $\mathbf{x}_k$:

$$\mathbf{G}_k^{int} = 2 \sum_j A_j \mathbf{n}_j \int_T \nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\, \phi_k(\mathbf{x}(\mathbf{u}))\, \mathrm{d}\mathbf{u}\,, \quad (30)$$

where the sum is on all the (completely) visible triangles containing vertex $\mathbf{x}_k$.

4.2.3 Term due the movement of the crepuscular cone

In this section we are then going to compute the additional term which appears when $\mathbf{x}_k$ is a horizon point on the occluding contour. In this case the energy variation during a surface movement is due to the volume created by the crepuscular cones. This movement is not affected by the dependency in $\mathbf{n}(\mathbf{x})$ since this is purely due to the visibility changes.

Let $H_{k,j}$ be the vector such that $[\mathbf{x}_k, \mathbf{x}_k + H_{k,j}]$ is the edge of the triangle $\mathcal{S}_j$ generating the horizon. The volume corresponding to the movement of the horizon can be parametrized by the points $\mathbf{y}(u, v)$ of the triangle $\{\mathbf{x}_k, \mathbf{x}_k + H_{k,j}, \mathbf{x}_k + v\ t\mathbf{V}_k\}$ generated by the movement of the horizon. More rigorously, it can be parametrized as the set of points $\mathbf{x}(u, v, r) = r\ \mathbf{y}(u, v)$ where $\mathbf{y}(u, v) = \mathbf{x}_k + u\ H_{k,j} + v\ t\mathbf{V}_k$; r corresponds to the depth of $\mathbf{x}$ in the view point direction; $r \in [1, \mathcal{T}_{(u,v)}]$, where it corresponds to the terminator of $\mathbf{x}_k$ when $r = \mathcal{T}_{(u,v)}$.
Let us note that $\mathbf{y}(u, v)$ depends on t; we emphasize this dependency by denoting $\mathbf{y}_t(u, v)$. By a change of variable, we get $\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0]) =$

$$\ldots - \frac{1}{2} \int_T \int_{r \in [1, \mathcal{T}_{(u,v)}]} \Big\{ f(r\ \mathbf{y}_t(u, v)) \, (H_{k,j} \wedge\ t\mathbf{V}_k) \cdot \frac{\mathbf{y}_t(u, v)}{|\mathbf{y}_t(u, v)|} r^2 \Big\} \mathrm{d}r\mathrm{d}u\mathrm{d}v\,, \quad (31)$$

where "..." corresponds to the part described in the previous paragraph. It follows that the limit of $\frac{\tilde{E}(\mathcal{S}[t]) - \tilde{E}(\mathcal{S}[0])}{t}$ when t tends to zero is the term in Equation (30) plus the following term

$$-\frac{1}{2} \int_T \int_{[1, \mathcal{T}_u]} f(r\ \mathbf{y}(u))\, (H_{k,j} \wedge \mathbf{V}_k) \cdot \frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} r^2 \mathrm{d}r\mathrm{d}u\mathrm{d}v\,, \quad (32)$$

where $\mathbf{y}(u) = \mathbf{x}_k + u\ H_{k,j}$ and $\mathcal{T}_u = \mathcal{T}_{(u,0)}$. Let us denote $\mathcal{L}(u)$ such that

$$\mathcal{L}(u) = \int_{[1, \mathcal{T}_u]} f(r\ \mathbf{y}(u)) r^2 \mathrm{d}r\,. \quad (33)$$

The right-hand part of equation (32) can be rewritten as

$$-\mathbf{V}_k \cdot \frac{1}{2} \int_{u \in [0,1]} \mathcal{L}(u) \left(\frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} \wedge H_{k,j} \right) (1 - u) \mathrm{d}u\,, \quad (34)$$

where $\mathcal{L}(u) = \int_{1,\mathcal{T}_u} f(r\ \mathbf{y}(u))r^2 dr$. Here we have $f(\mathbf{x}) = -\nabla \cdot \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x}))$. Here we can explicit $\mathcal{L}(u)$ as :

$$\begin{aligned}\mathcal{L}(u) &= -\int_{1,\mathcal{T}_u} \nabla g(r\mathbf{y}(u)) \cdot \frac{\mathbf{y}(u)}{\mathbf{y}(u)_z^3} \mathrm{d}r \\ &= -\big[g(T(\mathbf{y}(u))) - g(\mathbf{y}(u))\big] \frac{1}{\mathbf{y}(u)_z^3}\ ,\end{aligned} \tag{35}$$

where we have denoted $\mathbf{g}(\mathbf{x}) = g(\mathbf{x}, \mathbf{n}) \frac{\mathbf{x}}{\mathbf{x}_z^3}$ and $T(\mathbf{y}(u))$ is the point located behind $\mathbf{y}(u)$ in the direction of the viewpoint (i.e. its terminator point).

This gives the third part of the kth component of the gradient of $E(\mathcal{S})$ defined in Equation (5):

$$\begin{aligned}\mathbf{G}_k^{horiz} = \sum_{H_{k,j}} \frac{1}{2} \int_0^1 \Big\{ \big(g\left(T\left(\mathbf{y}(u)\right)\right) - g\left(\mathbf{y}(u)\right)\big) \\ \times \left(\frac{\mathbf{y}(u) \wedge H_{k,j}}{|\mathbf{y}(u)||\mathbf{y}(u)|_z^3} \right) (1-u) \Big\} \mathrm{d}u\ ,\end{aligned} \tag{36}$$

where the sum is on the edges containing $\mathbf{x}_k$ and which generate horizons. $H_{k,j}$ are the horizon edges around vertex $\mathbf{x}_k$.

4.2.4 Total Gradient Descent Flow

Finally, as explained in Section 2.2, the gradient descent flow depends on the choice of an inner product for the gradient. Here we use the L^2 gradient that allow to account for triangle area variations. As described previously and shown in [6, 12], changing the metric can result in more coherent gradient flows, but this study is out the scope of this paper. In this case the gradient descent flow for a point $\mathbf{x}_k$ corresponding to our energy functional defined on a visible domain is:

$$\begin{cases} \mathbf{x}_k(0) = \mathbf{x}_k^0\ , \\ \dfrac{d\mathbf{x}_k}{dt} = -\dfrac{1}{A_k} \left\{ \mathbf{G}_k^{int} + \mathbf{G}_k^{horiz} + \mathbf{G}_k^{norm} \right\}\ . \end{cases} \tag{37}$$

5 Applications

This section shows how the gradient descent framework presented in the previous sections can be used to perform mesh evolution. In particular several concrete examples that are commonly used in computer vision and graphics are presented. For more clarity and comparison purposes, the following examples use the L^2-gradient.

In the following examples, the mesh evolution algorithm that includes the remeshing and the topology changes is done using the Delaunay topology-adaptive meshes proposed by [33] written using CGAL [4].

We are performing minimization via gradient descent, which starts from some initialization. In most examples the initial shape is the visual hull of the scene, where we assume silhouettes can be easily computed or are given, or that stereoscopic segmentation [47] can be easily performed. It is also important to notice that the horizon term is performed only on existing contour generators, so that the topology has to be close to the final one, or that contour generators can be created thanks to the interior term. Also, since it is based on energy differences one may add more or less importance to it. In particular in the following experiments, we add a weight λ_H to control this amount, which is empirically determined. If λ_H is too big, the horizon term will make the contour generators of the surface oscillates around their corresponding occluding contours.

Finally, all firgures shown in this section are displayed using flat shading rendering.

5.1 Mean Curvature Flow

One of the most commonly used gradient flow is the mean curvature flow, that minimizes the surface area. This energy is often used to perform mesh smoothing, or more often as a smoothing energy term. The associated energy functional is simply

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathrm{d}s\ , \tag{38}$$

and its associated continuous gradient is $\nabla E = \kappa$, where κ is the surface's curvature.

However, one needs some approximations in order to apply it to discrete surface representations since notion of discrete curvature is not clear. On triangular meshes, different approximated flows have been proposed like laplacian approximation or the umbrella operator. Meyer et al [30] have computed the discrete gradient flow that mini-

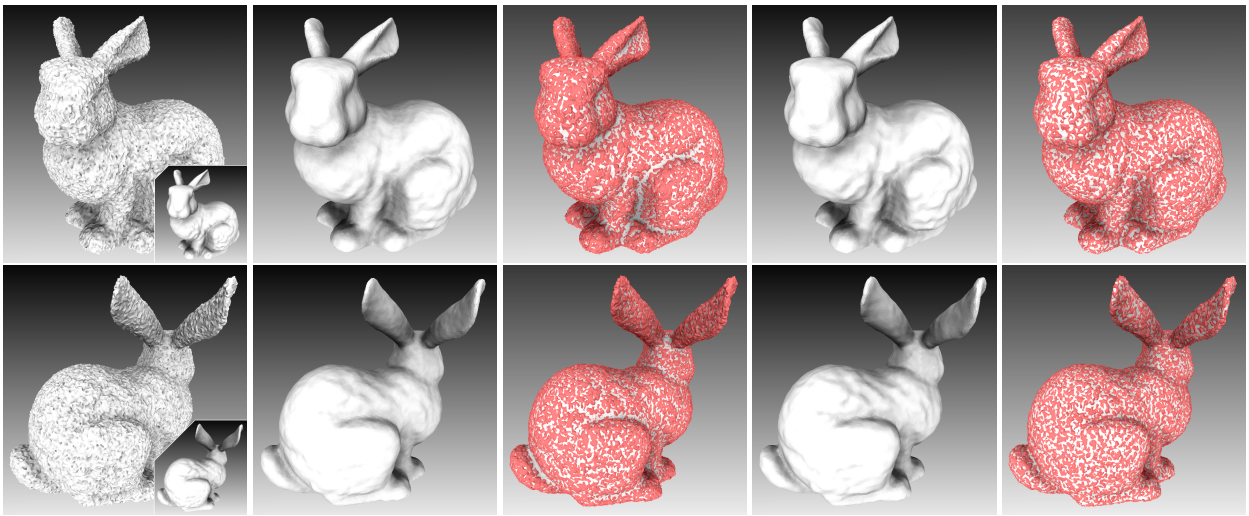


Figure 5: Evolution of smoothing algorithms on the Stanford Bunny data. From left to right: input noisy data; denoised mesh obtained using the mean curvature flow (MCF) of Section 5.1; MCF result versus the noisy input (in red); denoised mesh obtained using the normal smoothing algorithm (NS) of Section 5.2; NS result versus the noisy input (in red). Smoothing the normals help preserving details better than the mean curvature flow and does not over-smooth the result at concavities and convexities as much as the mean curvature flow. (Color Online)

mizes Equation (38) with respect to the triangle mesh representation. It is easy to show that the same result can be obtained using the presented approach. Following Section 2, we have

$$\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = \sum_k \mathbf{V}_k \cdot \sum_{j\in\mathcal{J}_k} \frac{1}{2}\mathbf{n}_j \wedge \mathbf{e}_{j,k} \,. \tag{39}$$

The evolution of one vertex k then follows

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k}\sum_{j\in\mathcal{J}_k}\frac{1}{2}\,\mathbf{e}_{j,k}\wedge\mathbf{n}_j \,, \tag{40}$$

where A_k is the area of the neighborhood $\mathcal{J}_k$ of vertex k. Note that this result is exactly the same as the results in [30], but is just expressed differently. The given formulation can be useful for applications where edges and surface normals have been previously computed. Figure 5 illustrates this algorithm on the Stanford Bunny data [1].

5.2 Normal Field Integration

One of the applications is to align a surface with respect to an external normal field. For instance, in 3D reconstruction, one recover the surface by integrating photometric normals [5, 20, 43]. Let $\mathbf{h}$ be a unit vector field in $\mathbb{R}^3$. Integrating this vector field $\mathbf{h}$ such that the surface normals $\mathbf{n}$ correspond to it involves minimizing the following functional

$$\begin{aligned} E(\mathcal{S}) &= \frac{1}{2}\int_{\mathcal{S}} |\mathbf{h}-\mathbf{n}|^2 \,\mathrm{d}s \\ &= \sum_j \int_{\mathcal{S}_j} (1-\mathbf{h}\cdot\mathbf{n}_j)\,\mathrm{d}s \,. \end{aligned} \tag{41}$$

As explained in Section 3, it is easy to show that the following gradient descent flow is:

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k}\sum_{j\in\mathcal{J}_k}\frac{1}{2}\,\mathbf{e}_{j,k}\wedge\int_T(\mathbf{n}_j-\mathbf{h})\,\mathrm{d}\mathbf{u} \,. \tag{42}$$

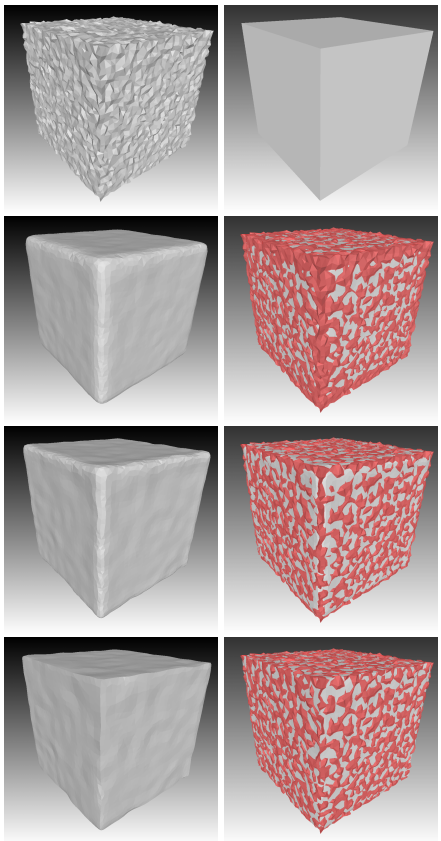


Figure 6: Evolution of smoothing algorithms on simple cube data. From top to bottom: noisy input and corresponding original surface; mean curvature flow smoothing; normal smoothing; median filtering. The red part corresponds to the input noisy mesh being displayed together with the results. (Color Online)

5.2.1 Normal Smoothing

As an example we show now that this approach can be easily applied to normal smoothing with efficiency, and is barely more time consuming than the mean curvature flow. Let $\mathbf{h}_j$ be the weighted average normal of the triangle j and its neighborhood N_j:

$$\mathbf{h}_j = \frac{\sum_{l \in N_j} \alpha_l \, \mathbf{n}_l}{\| \sum_{l \in N_j} \alpha_l \, \mathbf{n}_l \|} \,, \tag{43}$$

where the weights α_l can be chosen depending on the application. It could for instance be the area A_l of the triangle l, a Gaussian weight or more simply one can set $\alpha_l = 1$. Therefore the energy functional to be minimized can be expressed as:

$$E(\mathcal{S}) = \sum_j \int_{\mathcal{S}_j} (1 - \mathbf{h}_j \cdot \mathbf{n}_j) \, \mathrm{d}s \,, \tag{44}$$

where $\mathbf{h}_j$ is constant over the surface. Note that we cannot directly apply the results presented in Section 3.2 to energy (44) since it depends on the normal at several surface points. To minimize energy (44), we consider $\mathbf{h}_j$ to be fixed while updating the surface. In other words, we alternately update the surface and the vector field $\mathbf{h}$. The gradient descent flow corresponding to the normal smoothing energy with respect to the surface is

$$\frac{d\mathbf{x}_k}{dt} = \frac{1}{A_k} \sum_{j \in \mathcal{J}_k} \frac{1}{2} \, \mathbf{e}_{j,k} \wedge (\mathbf{n}_j - \mathbf{h}_j) \,. \tag{45}$$

Note the similarity of the above equation with the mean curvature flow (Equation (40)), and the fact that the gradient flow with respect to the surface is barely more time consuming. Results are shown in Figure 5, where the evolution is stopped once the noise is no longer visible. The figure also shows a comparison between the mean curvature flow. In particular, when displayed with the input noisy mesh, one may notice the different density of noise in the mesh (shown in red). Since this is a Gaussian noise, the quantity of noise should be uniform all over the surface, as it is almost the case for normal smoothing. However, proportion of the visible noisy mesh in the mean curvature flow example is much denser in the convex parts and disappears in concavities. While mean curvature flow is popular for surface regularization and smoothing, normal smoothing preserve details better and does not over-smooth as much as the mean curvature flow.

5.2.2 Median Filtering

Similarly, one may choose the external vector field $\mathbf{h}_j$ to be the median vector of the neighborhood N_j of $\mathcal{S}_j$ to perform median filtering on the mesh. $\mathbf{h}_j$ can be computed by first computing the spherical coordinates (θ_m, ϕ_m) of each normal $\mathbf{n}_m \in N_j$. Then by sorting the spherical

coordinates (θ_m, ϕ_m) along each component, one can obtain the median normal by getting back to cartesian coordinates. Results on a noisy cube and comparison with the previous smoothing algorithms are shown in Figure 6.

5.3 Multi-view Stereovision

In this example, we detail how the gradient defined in Section 4 can be used in three-dimensional surface reconstruction from images.

Multi-view stereo is the problem of recovering the shape of scenes using cameras, by assuming that matching or correspondences between different views can be obtained (by usually considering Lambertian constant brightness assumptions). Given a set of images of a scene taken from different camera positions, the goal is to reconstruct the shape $\mathcal{S}$, and optionally the appearance, of the object. Since it is the inverse problem of image rendering, this problem can be modeled in a Bayesian framework by minimizing the difference between the images of the reconstructed model and the observed ones, i.e. the reprojection error. The correct variational interpretation of the Bayesian analysis yields a minimization of some energy functional defined on the images [15, 23, 34, 39, 47] and requires the visibility of the surface points to be carefully accounted for. Also, only recently, some authors [15, 47] manage to rigorously and fully account for visibility in the optimization process. These works have proved the interests of doing so by showing that this forces the contours generators of the reconstructed surface to match with the apparent contours in the observed images. This makes the use of additional energy terms [19, 37, 44] unnecessary and significantly reduces the minimal surface bias.

The multi-view stereovision algorithm we propose here is based on the same modeling as the one proposed by [15]. As Gargallo et al. we fully account for visibility, and then acquire the same benefits (matching of the apparent contours and removing the minimal surface bias). The significant difference with respect to [15] relies on the surface representation. As [47], Gargallo et al. use the Eulerian formulation and implement their algorithm in the level-set framework. Here, we use a triangle mesh-based representation, as described in the previous sections. Below, we describe the modeling of the problem and then detail the exact gradient for our discrete representation obtained by using the results developed in Section 4.2.

5.3.1 Modeling of the Reprojection Error

In order to be able to compare the whole observed images (data) with the images generated by the model, it is crucial to define and model the background. Also, as shown by [15, 47], this allows us to be sure that the estimated foreground surface does not shrink to an empty set (which is the global optimum for most cost functionals used in other work). Moreover accounting for the background allows the contours generated by the recovered object to match with the occluding contours of the images. Whereas most of the previous work assumes that the background is known (e.g. simply modeling it as uniformly black, or by exploiting given silhouette images), here we also estimate the background images $B_i : \mathcal{I}_i \to \mathbb{R}^3$, under the single assumption that these images are smooth, similarly as [47]. Here i corresponds to the index of the camera and $\mathcal{I}_i$ is its image domain.

Now, let us assume that the scene surface $\mathcal{S}$ is Lambertian and the illumination static. Let $C : \mathcal{S} \to \mathbb{R}^3$ be the radiance function that associates colors to the points on the surface. Ideally, the color $I_i(\mathbf{p})$ observed at pixel $\mathbf{p}$ of image I_i should be equal to the color $C(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p}))$ of its backprojection $\pi_{i,\mathcal{S}}^{-1}(\mathbf{p})$ onto the surface or, in the case where $\mathbf{p} \notin \pi_i(S)$, to the color $B_i(\mathbf{p})$ of the same pixel on the background images (π_i denoting the projection associated with camera i). Thus, the reprojection error of the surface is

$$E_{data} = \frac{1}{2}\sum_i \left[\int_{\pi_i(\mathcal{S})} \left(I_i(\mathbf{p}) - C(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p}))\right)^2 \mathrm{d}\mathbf{p} + \int_{\mathcal{I}_i - \pi_i(\mathcal{S})} \left(I_i(\mathbf{p}) - B_i(\mathbf{p})\right)^2 \mathrm{d}\mathbf{p} \right] . \quad (46)$$

Finally, in order to well pose the problem, we use as a prior on $\mathcal{S}$ an additional smoothing area energy. As described previously, smoothing the normals gives a better prior than the commonly used mean curvature flow. It helps not to over-smooth the surface and to preserve geometric details better. The considered energy is

$$E_{R\mathcal{S}} = \int_{\mathcal{S}} (1 - \mathbf{h}(\mathbf{x}) \cdot \mathbf{n})\, \mathrm{d}s \,,$$

where $\mathbf{h}$ is the average normal of the considered surface point. We also assume that the background images are

smooth by adding the total variation term

$$E_{RB} = \sum_i \int_{\mathcal{I}_i} |\nabla B_i(\mathbf{p})| \mathrm{d}\mathbf{p}.$$

The total variation helps preserving edges and does not over-smooth the object boundaries. In practice, the background images can be identified before the surface optimization by giving the silhouettes or doing some stereoscopic segmentation.

5.3.2 Minimization of the Total Energy

For optimizing our total energy

$$E_{total} = E_{data} + \lambda_{\mathcal{S}} E_{R\mathcal{S}} + \lambda_B E_{RB} \,, \tag{47}$$

we perform gradient descents alternately with respect to B_i and $\mathcal{S}$.

The computation of the gradients with respect to B_i is classical since it is image-based [47]. For a fixed shape $\mathcal{S}$ and a fixed C, we have

$$\nabla E_{total}(B) = -\sum_i \left[(I_i - B_i)(1-h) + \lambda_B \nabla \cdot \frac{\nabla B_i}{\|\nabla B_i\|} \right] \tag{48}$$

where h is the characteristic function that indicates if $\mathbf{u}$ is covered by the projection of the surface $\mathcal{S}$ ($h(\mathbf{p}) = 1$) or not ($h(\mathbf{p}) = 0$, $\mathbf{p}$ being explained by the background); λ is the smoothness parameter; $C(\mathbf{x})$ is computed by taking the mean color of the projection in the image I_i where $\mathbf{x}$ is visible.

Let us now detail the gradient of E_{total} with respect to $\mathcal{S}$. Since E_{RB} do not depend on $\mathcal{S}$ and $E_{R\mathcal{S}}$ is classical [9, 47]), the main point is to compute the gradient of E_{data} with respect to $\mathcal{S}$. To apply the results presented in the previous sections, we first need to rewrite the energy E_{data} as an integral over only the visible surface. For simplicity of notation, we are going to give the gradient for a single camera and so we remove the dependency on i. For several cameras, the gradient will be the sum of the gradients associated with each camera.

As explained in the introduction, the first step, to be able to apply our previous results, is to rewrite the energy as an integral over the surface instead of the image. This change of variable implies the geometric model of the camera which we assume to be a pinhole perspective camera model in this work. It also involves adapting the measure on the surface [39] and in counting only the visible points [15, 34, 47]. This can be achieved by $d\mathbf{u} = -\frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3}\nu_{\mathcal{S}}(\mathbf{x})\mathrm{d}s$ where $\mathrm{d}s$ is the classical surface area measure and $\mathbf{x}_z$ is the depth of $\mathbf{x}$. Thus, by using the separation technique proposed by [47], the energy functional becomes (for a *single* image):

$$\begin{aligned} E_{data}(\mathcal{S}) = &-\int_{\mathcal{S}} g_I(\mathbf{x}) \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \nu_{\mathcal{S}}(\mathbf{x}) \mathrm{d}s \\ &\qquad + \int_{\mathcal{I}-\pi(\mathcal{S})} g_B(\mathbf{p}) \mathrm{d}\mathbf{p} \,, \\ = &-\int_{\mathcal{S}} [g_I(\mathbf{x}) - g_B(\pi(\mathbf{x}))] \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \nu_{\mathcal{S}}(\mathbf{x}) \mathrm{d}s \\ &\qquad + \int_{\mathcal{I}} g_B(\mathbf{p}) \mathrm{d}\mathbf{p} \,, \end{aligned} \tag{49}$$

where $g_I(\mathbf{x})$ is $\frac{1}{2}\left[I(\pi(\mathbf{x})) - C(\mathbf{x})\right]^2$ and $g_B(\mathbf{p})$ is $\frac{1}{2}\left[I(\mathbf{p}) - B(\mathbf{p}))\right]^2$. The right-hand term of equation (49) does not depend on $\mathcal{S}$, so in the following we intentionally omitted it as it does not contribute to the gradient expression (with respect to $\mathcal{S}$). Hence, denoting $g(\mathbf{x}) = g_I(\mathbf{x}) - g_B(\pi(\mathbf{x}))$ for convenience, the energy to be minimized with respect to $\mathcal{S}$ is

$$E_{data}(\mathcal{S}) = -\int_{\mathcal{S}} g(\mathbf{x}) \frac{\mathbf{x}\cdot\mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \nu_{\mathcal{S}}(\mathbf{x}) \mathrm{d}s. \tag{50}$$

Now, the energy functional $E_{data}(\mathcal{S})$ is of the form of Equation (5) with

$$\mathbf{g}(\mathbf{x}) = -\frac{g(\mathbf{x})}{\mathbf{x}_z^3}\mathbf{x}.$$

The gradient descent flow for the shape is then directly given by (37)

$$\begin{cases} \mathbf{x}_k(0) = \mathbf{x}_k^0 \,, \\ \dfrac{d\mathbf{x}_k}{dt} = -\dfrac{1}{A_k} \left\{ \mathbf{G}_k^{int} + \mathbf{G}_k^{horiz} \right\} \,, \end{cases} \tag{51}$$

where $\mathbf{G}_k^{int}$ and $\mathbf{G}_k^{horiz}$ are respectively obtained from Equations (30) and (36) where $\mathbf{g}(\mathbf{x})$ is replaced by $-\dfrac{g(\mathbf{x})}{\mathbf{x}_z^3}\mathbf{x}$.

5.3.3 Experiments for the multi-view stereovision application

We have implemented our algorithm using the Delaunay topology-adaptive meshes proposed by [33]. The visibility is computed using OpenGL Z-buffer. The evolution is done using a multi-resolution scheme and starting from the visual hull. Horizons are located using the changes of signs of the dot products of facet normals and viewpoint directions. The *terminator* error metric is computed using OpenGL Shader Language.

As in [15], we first reconstruct three uniformly colored balls arranged on a plane (20 images of resolution 640×480), see Figure 7. This way we ensure the importance of the *horizon* term as the color gradient is null over the surface except at the interfaces between objects in images (which correspond to object boundaries). Using only the *interior* term (Section 4.2.2), the surface shrinks due to the minimal bias. By using the horizon term only (given in Section 4.2.3), we correctly reconstruct and separate the balls, and occluding contours correctly reproject in the images. Then we tested our algorithm on synthetic

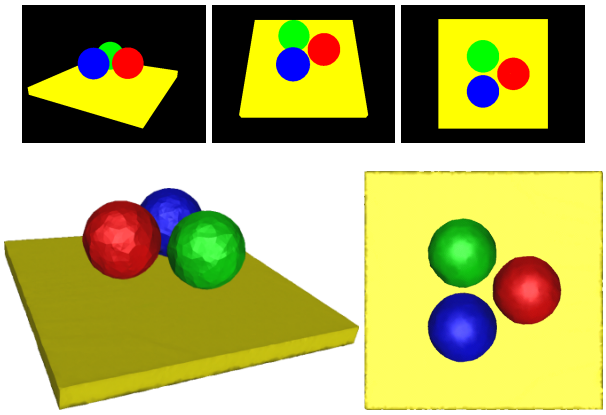


Figure 7: The balls sequence. Top row: 4 of 20 input images. Bottom row: results with the horizon term computed in Section 4.2.3 from different viewpoints.

Lambertian data for the Stanford dragon mesh (Figure 8) composed of 32 images of resolution 640×480. The result shows the correct reconstruction of the dragon, even though the texture is smooth and some parts in shadow are dark (See Table 1). Here the initial shape was a visual hull automatically computed from a stereoscopic segmentation algorithm [47].

	Dragon Diffuse images (Figure 8)
Acc. 95%	0.241mm
Comp. 0.5mm	98.3%

Table 1: Numerical evaluation of the proposed method for the dragon sequence that shows accuracy at 95% and completeness at 0.5mm following [36]. (compared to ground truth)

	templeSparseRing		dinoSparseRing	
	accu. (mm)	compl. (%)	accu. (mm)	compl. (%)
Gargallo [15]	1.05	81.9	0.76	90.7
Gargallo [14]	0.79	96.8	0.50	97.7
Our approach	0.73	95.9	0.89	93.9

Table 2: Results for the temple and dino datasets. For each dataset, accuracy and completeness scores are given. Results of Gargallo et al. are shown for comparison purposes.

Finally, we tested our method on the classical Dino (16 images of 640×480) and Temple (16 images of 640×480 images) datasets from the Middlebury repository (Figure 9). The results and a comparison with selected approaches that motivated our work is presented Table 2. The results of [14, 15] are done in the continuous domain using level set implementation. We refer to the Middlebury benchmark website [36] for evaluation with state-of-the-art reconstruction. We can see that our method is comparable to state-of-the-art, but the main contribution here results in giving a unified framework for photo-consistency optimization that correctly handle visibility using triangular meshes. Additional terms like ballooning forces or silhouettes terms can now be understood, by only dealing with the reprojection error criteria.

5.4 Lambertian 3D Reconstruction using Illumination

In the previous case, the illumination was not taken into account. In fact the estimated color of the surface com-

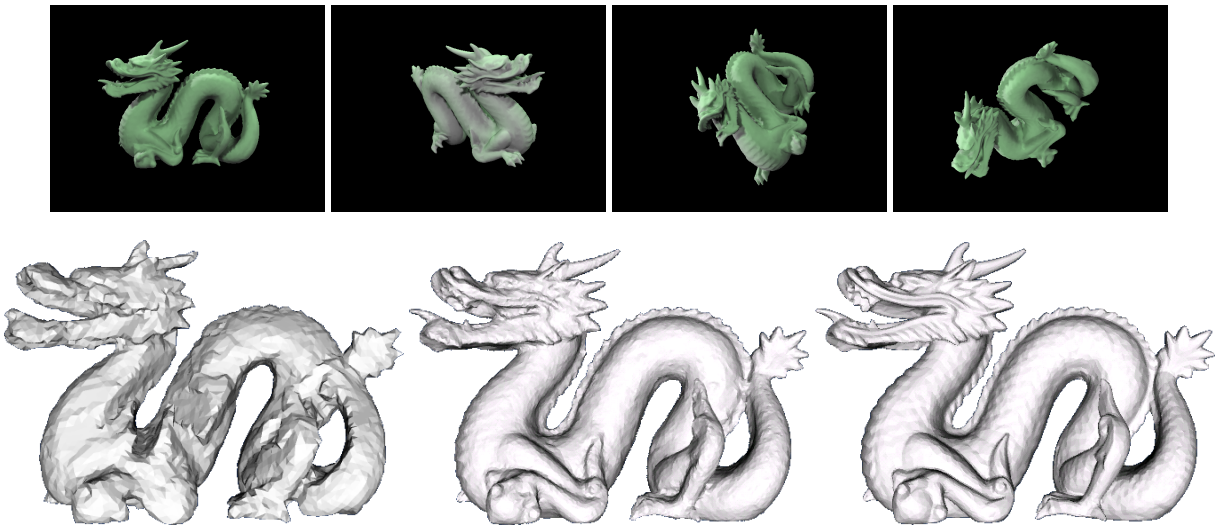


Figure 8: Synthesized dragon sequence. Top row: 4 of 32 diffuse input images. Bottom row: Initial shape; recovered shape by our algorithm; ground truth model.

pared to the input images was the estimated radiance of the object (which was supposed to be Lambertian). However, radiance contains shading, inter-reflections, cast shadows and other non-Lambertian phenomena. The illumination adds additional cues that can be used to estimate surface normals. Considering this, shading can be used in order to recover the geometry of textureless regions. This also allows to separate the surface albedo from the radiance and allows to do more realistic object relighting.

In this section, we only consider the case of Lambertian surfaces. In particular, this allows to consider the multi-view Shape From Shading problem (SFS) and the multi-view Photometric Stereo problem (PS). These problems consist of recovering the 3D shape of a scene by exploiting the information contained in the shading of the corresponding images. Basically in multi-view photometric stereo, the images are generated with varying lighting (typically, each point of the surface must be seen with three different lights). Whereas in multi-view shape from shading, the lighting is the same for all the images. The SFS problem is therefore less well-posed than the PS problem. It then needs some additional constraints. In SFS, we classically assume that the reflectance properties are homogeneous over the whole scene.

A solution to such problems would be a surface $\mathcal{S}$ such that the images generated from that surface are very similar to the observed images (i.e. the data). This naturally leads to formulate the problem as the minimization of an error measure between the observed and predicted values of pixels.

For simplicity, here we are going to consider only Lambertian scenes illuminated by point light sources. This work can nevertheless be extended to other parametric reflectance models and to more realistic lighting conditions for example as done in [21, 49]. For a point $\mathbf{x}$ of the surface $\mathcal{S}$, the radiance equation for the ith image is then

$$\begin{aligned} I_i(\pi_i(\mathbf{x})) &= \rho(\mathbf{x}) \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l,\mathcal{S}}^i(\mathbf{x})(\mathbf{n}(\mathbf{x}) \cdot \mathbf{l}_l^i(\mathbf{x})) + E_0 \right) \\ &= R(\mathbf{x}, \mathbf{n}(\mathbf{x}), \mathcal{S}) \ . \end{aligned} \tag{52}$$

Above L_l^i and $\mathbf{l}_l^i$ are respectively the light intensity color and the light direction of the lth light in the ith image. $\nu_{l,\mathcal{S}}^i(\mathbf{x})$ is the visibility of the lth light source of the ith image at point $\mathbf{x}$ according to $\mathcal{S}$. The additional term E_0

Figure 9: Dino and Temple sequence (input data courtesy of [36]). From left to right: 1 of 16 input images; estimated background images (scaled by 2 for visualization); estimated radiance; estimated mesh seen from a different viewpoint.

corresponds to the ambient lighting.
A natural energy functional to be considered can be written as :

$$E(\mathcal{S}) = \frac{1}{2}\sum_i \int_{\mathcal{I}_i} \left(I_i(\mathbf{p}) - R(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p}), \mathbf{n}(\pi_{i,\mathcal{S}}^{-1}(\mathbf{p})), \mathcal{S}) \right)^2 \mathrm{d}\mathbf{p} \tag{53}$$

see Jin et al. [21] for details.

To minimize this energy, we alternately minimize it with respects to the shape $\mathcal{S}$ and to the albedo ρ. For a fixed shape $\mathcal{S}$, the optimal albedo $\rho(\mathbf{x})$ is obtained using :

$$\rho(\mathbf{x}) = \frac{\sum_i I_i\left(\pi_i(\mathbf{x})\right) \nu_{\mathcal{S},i}(\mathbf{x})}{\sum_i \left(\sum_{l=1}^{n_L^i} L_l^i \nu_{l,\mathcal{S}}^i(\mathbf{x})(\mathbf{n}(\mathbf{x}) \cdot \mathbf{l}_l^i(\mathbf{x})) + E_0 \right) \nu_{\mathcal{S},i}(\mathbf{x})} \tag{54}$$

When we assume that the albedo is homogeneous, the denominators and numerators of the above equation have to be integrated on the whole surface $\mathcal{S}$. It has to be then multiplied by the adequate factor corresponding to the camera model; $\mathbf{n}(\mathbf{x}) \cdot \mathbf{x}/\mathbf{x}_z^3$ for a pinhole camera.
Now, let us fix the albedo and optimize energy (53) with respects to the shape. To simplify, we are going to neglect the variations of the visibility of the light sources $\nu_{l,\mathcal{S}}(\mathbf{x})$ when the shape is deforming. These variations are null almost everywhere. They are Dirac functions with a support restricted to the shadow boundaries. In other words, we are neglecting shadow information. Practically in the following, this is equivalent to assuming that $R(\mathbf{x}, \mathbf{n}, \mathcal{S})$ does not depend on $\mathcal{S}$.
In other respects, as previously done in other applications, let us note that the gradient of the energy (53) is the sum of the gradients associated with each one of the images. In the sequel we only compute the gradient associated with one image. By assuming that the camera is a pinhole, we can rewrite this energy as:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \mathbf{g}(\mathbf{x}, \mathbf{n}(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\, \nu_{\mathcal{S}}(\mathbf{x})\, \mathrm{d}s\,, \tag{55}$$

where

$$\mathbf{g}(\mathbf{x}, \mathbf{n}) = -\frac{1}{2}\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n})\right)^2 \frac{\mathbf{x}}{\mathbf{x}_z^3} \tag{56}$$

and where

$$R(\mathbf{x}, \mathbf{n}) = \rho(\mathbf{x}) \left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x})(\mathbf{n} \cdot \mathbf{l}_l(\mathbf{x})) + E_0 \right)\,. \tag{57}$$

We can then directly use the results of Section 4. The gradient is split into three parts: one for the term depending on normals, one for the interior term, and a last one for the horizon term.
For the term depending on normals, we have :

$$\mathbf{G}_k^{norm} = -\sum_{j \in \mathcal{J}_k} \mathbf{e}_{j,k} \wedge \int_T P_{n_j \top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)^T \mathbf{n}_j)\, \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))\mathrm{d}\mathbf{u} \tag{58}$$

where $P_{n_j \top}(D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^T \mathbf{n}_j)$ is the projection of $D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^T \mathbf{n}_j$ on the tangent plane of the surface. It can be re-written as $D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^T \mathbf{n}_j - \left(\mathbf{n}_j \cdot D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^T \mathbf{n}_j\right) \mathbf{n}_j$. Here we have

$$D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j) = \rho(\mathbf{x})\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n}_j)\right) \times \frac{\mathbf{x}}{\mathbf{x}_z^3} \left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) \mathbf{l}_l(\mathbf{x})^T \right)\,; \tag{59}$$

$\frac{\mathbf{x}}{\mathbf{x}_z^3}\left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) \mathbf{l}_l(\mathbf{x})^T\right)$ being a 3 matrix, so

$$D_{\mathbf{n}}\mathbf{g}(\mathbf{x}, \mathbf{n}_j)^T \mathbf{n}_j = \rho(\mathbf{x})\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n}_j)\right) \times \left(\frac{\mathbf{x}}{\mathbf{x}_z^3} \cdot \mathbf{n}_j\right) \left(\sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) \mathbf{l}_l(\mathbf{x}) \right)\,. \tag{60}$$

The kth component of the *interior* term is:

$$\mathbf{G}_k^{int} = \sum_j A_j \mathbf{n}_j \int_T \nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\, \phi_k(\mathbf{u})\, \mathrm{d}\mathbf{u}\,, \tag{61}$$

where the sum is on the set of the (completely) *visible* triangles $\mathcal{S}_j$ containing the vertex $\mathbf{x}_k$, and where

$$\nabla \cdot \mathbf{g}(\mathbf{x}(\mathbf{u}), \mathbf{n}_j) = -\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n})\right) \times \left(D\pi(\mathbf{x})^T \nabla I(\mathbf{x}) - \nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j)\right) \cdot \frac{\mathbf{x}}{\mathbf{x}_z^3}\,. \tag{62}$$

Above, all the terms are explicit at the exception of $\nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j)$. In fact $\nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j) = \nabla\rho(\mathbf{x})\mathrm{L}(\mathbf{x}, \mathbf{n}_j) + \rho(\mathbf{x})\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x}, \mathbf{n}_j)$ where we denote $\mathrm{L}(\mathbf{x}, \mathbf{n}_j) = \sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x})(\mathbf{n}_j \cdot \mathbf{l}_l(\mathbf{x})) + E_0$. The computation of the term $\nabla_{\mathbf{x}} R(\mathbf{x}, \mathbf{n}_j)$ and $\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x}, \mathbf{n}_j)$ are detailed and discussed in Appendix A.3.

The kth component of the *Horizon* term is:

$$\mathbf{G}_k^{horiz} = -\sum_{H_{k,j}} \frac{1}{2} \int_0^1 \mathcal{L}(u) \left(\frac{\mathbf{y}(u)}{|\mathbf{y}(u)|} \wedge H_{k,j} \right) (1-u) \mathrm{d}u \,, \tag{63}$$

where $\mathcal{L}(u) = -\big[h(T(\mathbf{y}(u))) - h(\mathbf{y}(u))\big] \frac{1}{\mathbf{y}(u)_z^3}$ with $h = \frac{1}{2}\left(I(\pi(\mathbf{x})) - R(\mathbf{x}, \mathbf{n}(\mathbf{x}))\right)^2$ and $T(\mathbf{y}(u))$ is the *terminator* point of $\mathbf{y}(u)$.

5.5 Multi-view Normal Integration

In this section, we present an application for integrating surfaces from multiple normal maps like for instance the one of Chang et al.[5] developed in the level sets framework. Such normals can for instance be obtained via photometric stereo that uses a single fixed camera and a moving light source [20, 43]. Having different illumination conditions for one particular view-point allows to estimate the surface normals. By integrating this normal field, it is possible to recover the 3D geometry of the scene. This can be done using the previously described method for normal field integration. However, since photometric stereo is a vision-based application that allows to recover normals for each pixel in the image, the energy functional is based on camera modeling and therefore the energy can be expressed as a reprojection error functional. The gradient descent corresponding to this problem then directly follows the approach presented in Section 4.

The problem can be solved by minimizing the following energy functional:

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{I}} \frac{1}{2} \left(N_i(\mathbf{p}) - \mathbf{n}(\pi_{\mathcal{S}}^{-1}(\mathbf{p})) \right)^2 \mathrm{d}\mathbf{p} \,, \tag{64}$$

where $N(\mathbf{p})$ is the normal in input image and $\mathbf{n}(\mathbf{x})$ is the normal of the surface $\mathcal{S}$ at point $\mathbf{x}$. As the norms of N and $\mathbf{n}$ are equal to 1, for simplicity one can rewrite Equation (64) for a *single* image as

$$E(\mathcal{S}) = \int_{\mathcal{I}} \left(1 - N(\mathbf{p}) \cdot \mathbf{n}(\pi_{\mathcal{S}}^{-1})\right) \mathrm{d}\mathbf{p} \,. \tag{65}$$

Rewriting it as an integral over the visible surface, we have:

$$E(\mathcal{S}) = \int_{\mathcal{S}} \left(1 - N(\pi(\mathbf{x})) \cdot \mathbf{n}(\mathbf{x})\right) \frac{\mathbf{x} \cdot \mathbf{n}(\mathbf{x})}{\mathbf{x}_z^3} \nu_{\mathcal{S}}(\mathbf{x}) \mathrm{d}s \,, \tag{66}$$

which has a similar form as the one used previously for the reprojection error with $g(\mathbf{x}, \mathbf{n}) = (1 - N \cdot \mathbf{n})$. This way one can use previous results when the energy functional also depends on the normal. The differential of the energy with respect to a vertex $\mathbf{x}_k$ for the term due to the normal is

$$\mathbf{V}_k \cdot \sum_j \mathbf{e}_{j,k} \wedge \int_{T_j} \Big\{ \Big(N - (N \cdot \mathbf{n}_j)\, \mathbf{n}_j \Big) \frac{\mathbf{x} \cdot \mathbf{n}_j}{\mathbf{x}_z^3} \nu_{\mathcal{S}}(\mathbf{x})\, \phi_k(\mathbf{x}(\mathbf{u})) \Big\} \mathrm{d}\mathbf{u} \,. \tag{67}$$

Then, to get the complete gradient, one has to sum (the gradient corresponding to Equation (67)) with the term due to the differential of a quantity integrated over a visible volume (containing the *interior* term and the *horizon* term). The *interior* term is null because on the triangle $\nabla \cdot \mathbf{n}_j = 0$ and because $(\nabla N) \cdot \mathbf{x} = 0$. The Horizon term is:

$$\mathbf{V}_k \cdot \sum_{H_{k,j}} \frac{1}{2} \int_0^1 \Big\{ \Big(\mathbf{n}_j - \mathbf{n}(T(\mathbf{y}(u))) \Big) \cdot N(u) \frac{\mathbf{y}(u) \wedge H_{k,j}}{|\mathbf{y}(u)|\, [\mathbf{y}(u)]_z^3} (1-u) \Big\} \mathrm{d}u \,, \tag{68}$$

where $T(\mathbf{y}(u))$ is the *terminator* of the current point $\mathbf{y}(u)$ (located behind $\mathbf{y}(u)$ in the view point direction). Note that compared to the gradient in the continuous case, we have here a lower complexity (we are missing the divergence operator and have instead the vectorial product with the opposite edge of the triangle). It is more natural to implement on triangular meshes than the previous continuous case [14] :

$$\nabla \Big((N - (N \cdot \mathbf{n})\mathbf{n}) \frac{\mathbf{x} \cdot \mathbf{n}}{\mathbf{x}_z^3} + (N - \mathbf{n})^2 \frac{\mathbf{x}}{\mathbf{x}_z^3} \Big) \nu_{\mathcal{S}} + \Big((N - \mathbf{n})^2 - (N - \mathbf{n}')^2 \Big) \frac{\mathbf{x}^t \nabla \mathbf{n} \mathbf{x}}{\mathbf{x}_z^3} \delta(\mathbf{x} \cdot \mathbf{n}) \nu_{\mathcal{S}} \,. \tag{69}$$

As described previously, this can be extended to multi-view photometric stereo methods, where normals are estimated using reflectance and lighting conditions. Then it is possible to integrate this normal field estimated for each image pixel in order to recover the full 3D shape [5, 20, 43]. In the following, we illustrate the approach

with different example, where we also add a smoothness term . The corresponding energy is:

$$E_{RS} = \int_{\mathcal{S}} (1 - \mathbf{h}(\mathbf{x}) \cdot \mathbf{n}) \, \mathrm{d}s \, ,$$

where $\mathbf{h}(\mathbf{x})$ corresponds to the mean of all the normals viewed from each camera at point $\mathbf{x}$. The corresponding gradient is a straight forward application of Section 5.2.

Figure 10, and 12 illustrates our method on synthetic examples for multi-view normal field integration. First, we tested the multiview normal integration algorithm on the simple Ellipse dataset. By using only the term depending on the normal, the surface shrinks. The horizon term allows to constraint the surface such that it matches the image contours. It naturally gives boundary conditions for the normal integration and allows to start from surfaces that does not fully contain the object of interest.

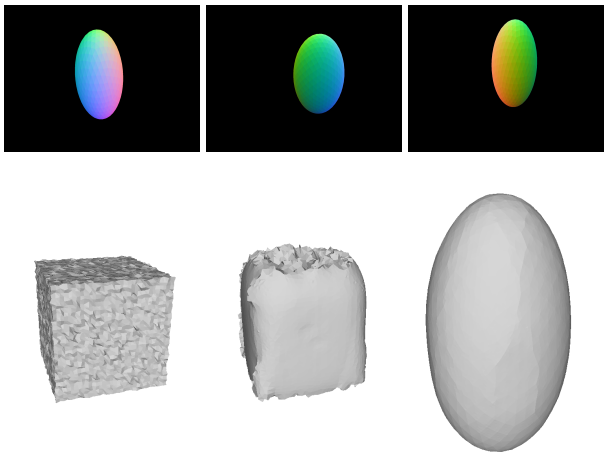


Figure 10: The ellipse sequence. Top row: 3 of 24 input images showing normal maps of the object of interest; Bottom row: Initial surface; Intermediate result during the evolution; Reconstructed surface.

The second experiment shown in Figure illustrates the approach on a CAD designed mesh. The original mesh as twice more vertices and triangles than the reconstructed one. Moreover, since we use a coherent gradient descent flow with respect to the mesh representation, we do not assume normal velocity like in [3, 10], making vertices

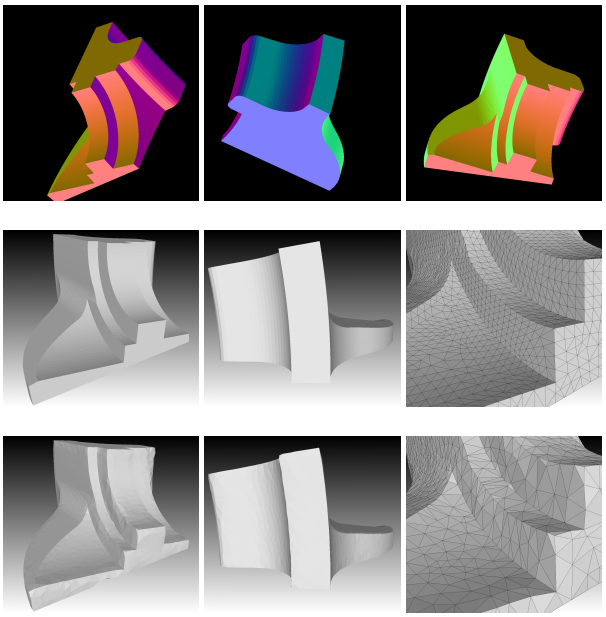


Figure 11: The Fandisk sequence. From Top to Bottom rows: 3 of 24 input images showing normal maps of the object of interest; original CAD model; reconstructed mesh. The last column shows details of the meshes with the associated triangulation. It shows that the coherent gradient flow makes triangle edges match with the data.

move to appropriate locations. This makes the recovered triangles nicely matches image edges even though the mesh resolution is not very high. Our method can then be used to reconstruct surfaces with sharp edges which is, as far as we know, not possible using implicit surface representations.

The third experiment (Figure 12) shows the efficiency of the proposed method for handling complex surfaces. The initial surface is the visual hull and a coarse to fine approach is used for the evolution. Details of the result are well recovered, and the final shape is very similar to the ground truth even though the input images have low resolution (640×480). Using photometric information, our method can then be used in order to obtain high quality meshes.

In order to compare ours results to state-of-the-art methods, we tested our approach on the dynamic photo-

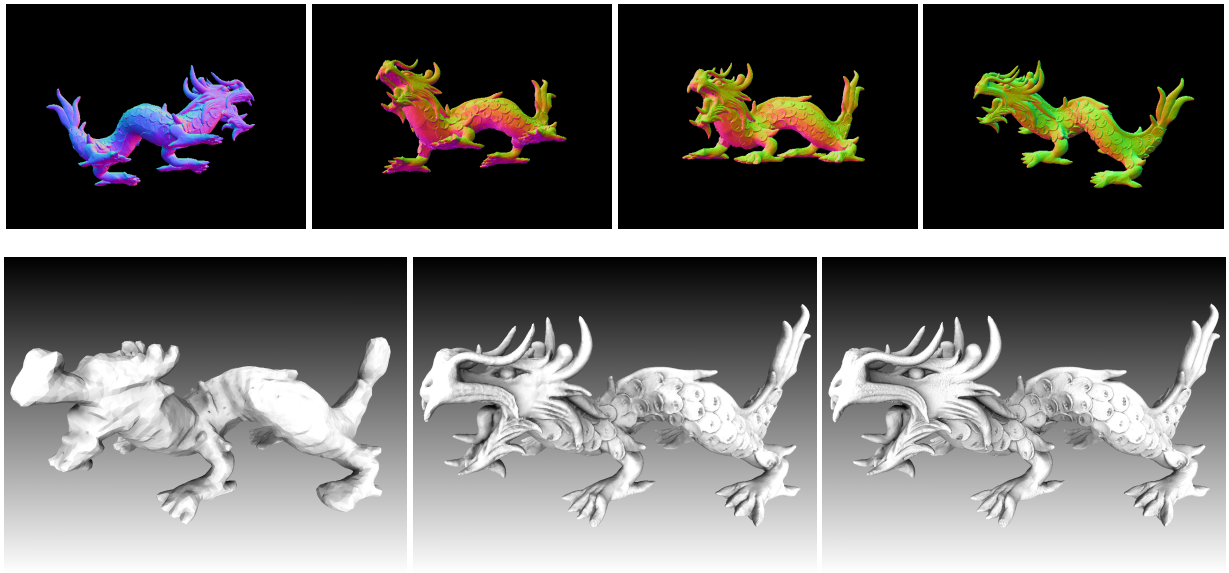


Figure 12: The dragon sequence (mesh obtained from Stanford repository [1]). Top row: 4 of 24 input images (640×480) showing normal maps of the object of interest; Bottom row: Initial surface; Reconstructed surface; Ground truth shape. Rendering is performed using ambient vertex occlusion and flat shading so that it displays the surface without missing any details.

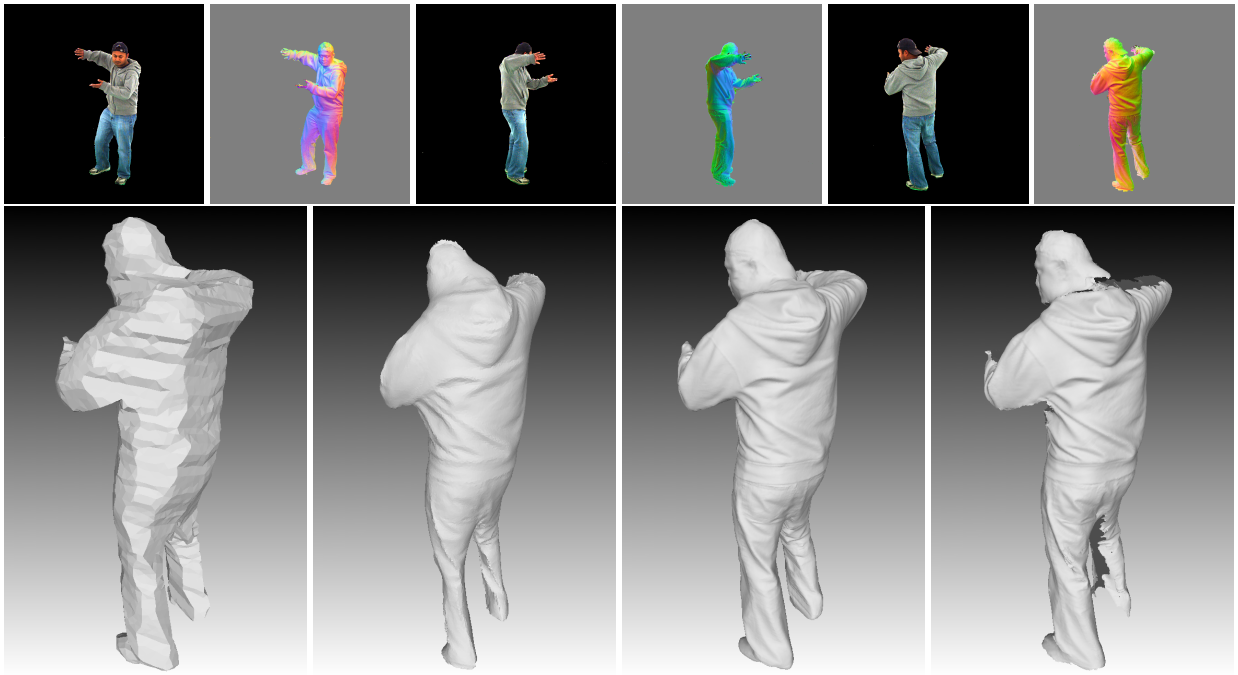


Figure 13: MIT Sequence (Courtesy of [43]). Top row: 3 of 8 input images and normals (1024×1024). Bottom row, from left to right: visual hull; result using Section 5.5 *wihtout* using the horizon term; result using Section 5.5 *with* the horizon term; result of [43].

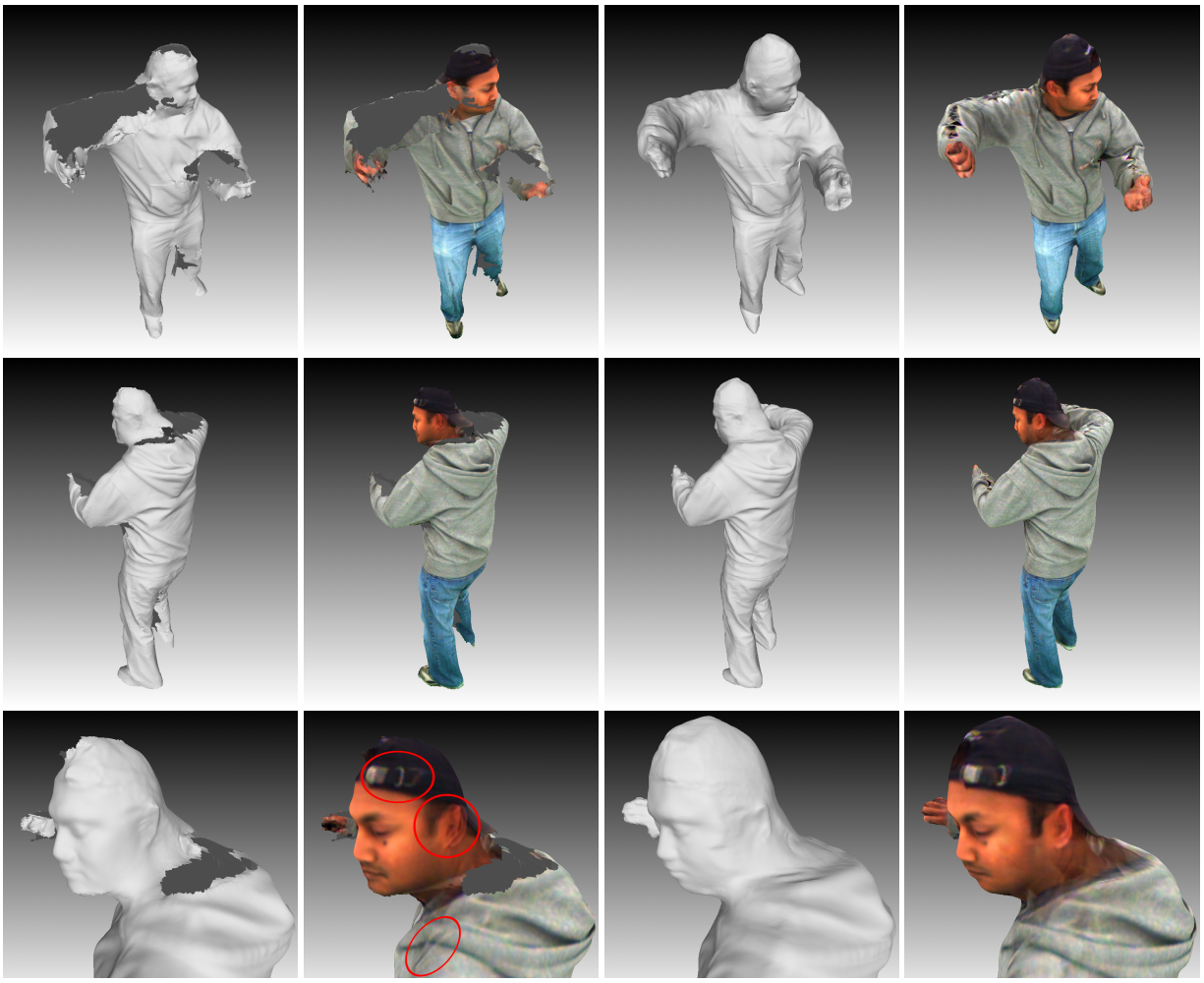


Figure 14: Results and comparison with [43]. From left to right: final shape of [43]; corresponding mean texture from visible cameras; result with the proposed approach; corresponding mean texture from visible cameras.

metric stereo dataset provided by Vlasic et al. [43]. It is composed of 8 images associated with 8 normal maps. The presented approach can directly be applied to their dataset by performing normal integration. Figure 13 illustrates those results. One of the images shows the shape obtained without using the *horizon* term. In this case the surface shrinks toward the empty set. The result on the right is obtained using the same flow plus the one of the *horizon* term, which yields the expected result. Integrating the normal field gives good high frequency and details, but is poor for low frequency due to the integration. In these conditions, mixing multiview stereo and multiview normal field integration will provide powerful 3D reconstruction algorithms [31, 43]. Figure 14 shows the results obtained by [43] along with ours. In their paper, the authors compute several normal maps from each view, and then register and merge the different integrations in order to obtain the final mesh. In this context they have troubles in recovering parts where the normal maps contain occluding contours. Since our approach is surface-based, we can better exploit the multi-view system in the reconstruction process and the 3D position of the surface is more accurate even though both methods nicely recover shape details (note that there are *only* 8 images). In fact, Figure 14 also shows the textured meshes obtained by reprojecting camera images onto the mesh. This emphasize the fact that even though the recovered surface of [43] visually looks really nice, they suffer from the integration bias and registration errors which leads to slightly incorrect 3D positions, as well as incomplete surface recovery. For example, the two images seeing the right ear reproject in different locations, creating a non-coherent textured surface - other problems are shown in red. This might result in wrong visual artifacts for relighting purposes. In contrast, our approach naturally takes advantage of the multi-view information. Also, our approach is purely image-based and does not use pre or post-processing such as re-estimating (and smoothing) the normal maps or performing hole filling like in [43]. Some part like the cap are not well recovered in our case, mainly due to noise and missing normals in the input images, but also by the fact we use a closed surface.

5.6 Discussion

The presented approach offers a general framework to solve different vision reconstruction problems using deformable meshes. However the way it is optimized might be improved since it uses simple L^2 gradient descent. One may change the gradient metric, make the functional convex, change the optimization algorithm or simply define a more robust cost measure to improve robustness or speed. In particular the initialization is important here, and there might be cases were the minimization fails because of local minima, even though coarse-to-fine approach (in both images and mesh resolution) significantly help the algorithm. A study of possible improvements is out the scope of this work but this paper still provides tools and inspiration for that (for example changing the gradient metric for deformable meshes is a straight forward extension of [12], and [45] already proposed a global error metric based on cross-correlations for triangle meshes based on [8, 34]). This work focuses more on the modeling part, by explaining the shape directly from images, rather than reconstructing surfaces under constraints (i.e. processing time, fast convergence, etc. . .). The paper however shows that it can be successfully applied for mesh refinement in a variety of cases, including geometric flows that depends on the surface's normal and/or visibility. Moreover one may adapt its error metric $\mathbf{g}$ for concerned applications - for example adding volumetric flows such as silhouettes constraints or edge attachment weight as done in geodesic active contour methods. In all cases adapting the metric is straight forward and its minimization via L^2 gradient descent is a direct application of the presented approach. Finally, in this work we used deformable meshes which imply remeshing (in particular for topology changes) during the optimization. In this context, adding or removing points might slightly change the objective functional. One additional improvement would be to change the remeshing algorithm in order to completely ensure the spatial consistency in the optimization process, but this is out of the scope of this paper and is still an open research area.

6 Conclusion

In this paper we compute the shape gradient of general energy functionals which account for normals and visi-

bility changes and which embody a number of energies used in mesh processing and computer vision. Gradient computation is done directly with respect to the discrete representation of the surface based on triangular meshes. This allows for coherent gradient flows that tend to place the mesh vertices to their correct locations and make triangle edges match with the data. To illustrate the presented approach and show the advantage of having a coherent gradient flow, we apply our results to several applications.

In particular we presented a mesh evolution technique for 3D vision problems, when the cost functional depends on an image based score and a camera model. For instance, we show the multiview stereo problem based on the *discrete* representation. Contrary to previous works, during the evolution, *we correctly deal with visibility changes* by expressing the exact gradient of the reprojection error functional. In particular, exactly as in the continuous case [15], this forces the contour generators of the surface to appear at their correct location in the images and reduces the minimal surface bias from which some variational methods suffer.

Acknowledgements The authors want to thank J.-P. Pons for sharing his implementation of the topologically-adaptive deformable meshes and for the useful discussions and help; S.M. Seitz, B. Curless, J. Diebel, D. Scharstein and R. Szeliski for the temple and dino datasets and evaluations; D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz and W. Matusik for the dynamic multi-view photometric stereo data; the Stanford repository team for their data; P. Gargallo for the usefull discussions and help and the ball dataset; and the anonymous reviewers for their work that helped to improve the quality of this paper. This work was supported by the Agence Nationale pour la Recherche within the Flamenco project (Grant ANR-06-MDCA-007).

References

[1] The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3dscanrep/.

[2] M. Bertalmio, G. Sapiro, Li-Tien Cheng, and S. Osher. Variational problems and pdes on implicit surfaces. *Variational and Level Set Methods in Computer Vision*, 2001.

[3] N. Birkbeck, D. Cobzas, P. Sturm, and M. Jägersand. Variational shape and reflectance estimation under changing light and viewpoints. In *Proceedings of European Conference on Computer Vision*, volume 1, pages 536–549, 2006.

[4] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org/.

[5] J. Y. Chang, K. M. Lee, and S. U. Lee. Multiview normal field integration using level set methods. In *IEEE Conference in Computer Vision and Pattern Recognition*. IEEE Computer Society, 2007.

[6] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras. Generalized gradients: Priors on minimization flows. *International Journal of Computer Vision*, 73(3):325–344, Jul 2007.

[7] É. Debreuve, M. Gastaud, M. Barlaud, and G. Aubert. Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. *Journal of Mathematical Imaging and Vision*, 2007.

[8] A. Delaunoy, E. Prados, P. Gargallo, J.-P. Pons, and P. Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *British Machine and Vision Conference, Leeds, UK*, 2008.

[9] M. Desbrun, M. Meyerm, P. Schröder, and A. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99*, pages 317–324, 1999.

[10] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In *European Conference on Computer Vision*, pages 238–251, Prague, Czech Republic, 2004.

[11] G. Dziuk and C.M. Elliott. Finite elements on evolving surfaces. *IMA journal of numerical analysis*, 27(2):262–292, 2007.

[12] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, and M. Desbrun. Generalized surface flows for mesh processing. In *Eurographics Symposium on Geometry Processing*, 2007.

[13] O.D. Faugeras and R. Keriven. Variational-principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE TIP*, 7(3):336–344, 1998.

[14] P. Gargallo. *Contributions to the Bayesian approach to Multi-view Stereo*. PhD thesis, Institut National Polytechique de Grenoble, France, February 2008.

[15] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *Proceedings of the International Conference on Computer Vision, Rio de Janeiro, Brazil*. IEEE Computer Society Press, 2007.

[16] B. Goldlucke, I. Ihrke, C. Linz, and M. Magnor. Weighted minimal hypersurface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1194–1208, 2007.

[17] B. Goldlücke and M. A. Magnor. Weighted minimal hypersurfaces and their applications in computer vision. In T. Pajdla and J. Matas, editors, *European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 366–378. Springer, 2004.

[18] R. Gupta and R. I. Hartley. Linear pushbroom cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):963–975, 1997.

[19] C. Hernandez and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004.

[20] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[21] H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 76(3), March 2008.

[22] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005.

[23] H. Jin, A.J. Yezzi, and S. Soatto. Variational multiframe stereo in the presence of specular reflections. In *3DPVT*, pages 626–630, 2002.

[24] K. Kolev, D. Cremers Integration of multiview stereo and silhouettes via convex functionals on convex domains In *European Conference on Computer Vision*, oct 2008.

[25] K. Kolev, D. Cremers Continuous ratio optimization via convex relaxation with applications 3D multiview reconstruction In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[26] K. Kolev, M. Klodt, T. Brox, D. Cremers Continuous global optimization in multiview 3D reconstruction In *International Journal of Computer Vision*, 2009.

[27] K. Kolev, T. Pock, D. Cremers Anisotropic Minimal Surfaces Integrating Photoconsistency and Normal Information for Multiview Stereo In *European Conference on Computer Vision*, 2010.

[28] Kolmogorov and Zabih. What energy functions can be minimized via graph cuts. *IEEETPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[29] P. Labatut, R. Keriven, and J.-P. Pons. Fast level set multi-view stereo on graphics hardware. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pages 774–781, Washington, DC, USA, 2006. IEEE Computer Society.

[30] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. 2002.

[31] D. Nehab, S. Rusinkiewicz, J. Davis and R. Ramamoorthi Efficiently combining positions and normals for precise 3D geometry In *SIGGRAPH*, 536–543, 2005.

[32] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[33] J.-P. Pons and J.-D. Boissonnat. A lagrangian approach to dynamic interfaces through kinetic triangulation of the ambient space. *Computer Graphics Forum*, 26(2):227–239, Aug 2007.

[34] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 822–827, San Diego, USA, Jun 2005.

[35] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision*, 72(2):179–193, Apr 2007.

[36] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2006.

[37] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. In *IEEE International Conference on Computer Vision*, pages 349–356, 2005.

[38] G. Slabaugh and G. Unal. Active polyhedron: surface evolution theory applied to deformable meshes. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:84–91 vol. 2, 2005.

[39] S. Soatto, A. J. Yezzi, and H. Jin. Tales of shape and radiance in multi-view stereo. In *IEEE International Conference on Computer Vision*, pages 974–981, 2003.

[40] J. E. Solem, H. Aanaes, and A. Heyden. A variational analysis of shape from specularities using sparse data. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 26–33, Washington, DC, USA, 2004. IEEE Computer Society.

[41] J. E. Solem and A. Heyden. Reconstructing open surfaces from image data. *Int. J. Comput. Vision*, 69(3):267–275, 2006.

[42] J. E. Solem and N. Chr. Overgaard. A geometric formulation of gradient descent for variational problems with moving surfaces. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space 2005*, volume 3459 of *LNCS*, pages 419–430. Springer Verlag, 2005.

[43] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz and W. Matusik Dynamic Shape Capture using Multi-View Photometric Stereo In *ACM Transactions on Graphics*, 28(5):174, 2009.

[44] G. Vogiatzis, C. Hernández Esteban, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12):2241–2246, 2007.

[45] H. Vu, R. Keriven, P. Labatut, and J.-P Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, Jun 2009.

[46] R. T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.

[47] A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1):31–43, 2003.

[48] A.J. Yezzi and S. Soatto. Stereoscopic segmentation. *IEEE International Conference on Computer Vision*, 2001.

[49] K-J. Yoon, E. Prados, and P. Sturm. Joint estimation of shape and reflectance using multiple images with known illumination conditions. *International Journal of Computer Vision*, 2009. to appear.

[50] T. Yu, N. Xu, and N. Ahuja. Shape and view independent reflectance map from multiple views. *International Journal of Computer Vision*, 73(2):123–138, 2007.

[51] A. Zaharescu, E. Boyer, and R.P. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Proceedings Asian Conference on Computer Vision*, Tokyo, Japan, November 2007.

A Computation details

A.1 Expression of $A'_j[0]$

We have to explicit $A'_j[0]$ linearly in function of the $\mathbf{V}$. The area $A_j[t]$ of a triangle $\mathcal{S}_j[t]$ is $A_j[t] = \frac{1}{2} \parallel \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]} \parallel$ where $\mathbf{x}_k[t],\ \mathbf{x}_{k1}[t],\ \mathbf{x}_{k2}[t]$ are the vertices of triangle i at time t. For more convinience, we express the squared area A_j^2 to avoid squared root while computing the differential at $t = 0$. Then we have :

$$A_j^2[t] = \frac{1}{4}\left(\overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]}\right)\cdot\left(\overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k1}[t]} \wedge \overrightarrow{\mathbf{x}_k[t]\mathbf{x}_{k2}[t]}\right)$$

$$\frac{d}{dt}A_j^2[t]\Big|_{t=0} = \frac{1}{2}\ \left(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\right) \cdot \left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k + \overrightarrow{\mathbf{x}_{k2}\mathbf{x}_k} \wedge \mathbf{V}_{k1} + \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \mathbf{V}_{k2}\right).$$

Using $\frac{d}{dt}A_j^2[t]\Big|_{t=0} = 2\,A_j[0]\,\frac{d}{dt}A_j[t]\Big|_{t=0}$, we get

$$A'_j[0] = \sum_{k\in K_j} \mathbf{V}_k \cdot \left(\frac{1}{2}\mathbf{n}_j \wedge \mathbf{e}_{j,k}\right) . \tag{70}$$

If we move only one vertex at once (meaning V_{k1} & V_{k2} are null for vertex k), we have :

$$A'_j[0] = \frac{1}{2}\ \left(\mathbf{n}_j \wedge \overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}\right) \cdot \mathbf{V}_k . \tag{71}$$

A.2 Expression of $\mathbf{n}'_j[0]$

We have to explicit $\mathbf{n}'_j[0]$ linearly in function of the $\mathbf{V}$. Considering $\mathbf{n}_j = \frac{\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}}{2\,A_j}$, we have :

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j[0]^2}\Big(\left(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\right)'[0]\,A_j[0] - \left(\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \overrightarrow{\mathbf{x}_k\mathbf{x}_{k2}}\right)A'_j[0]\Big).$$

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j}\Big(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}}\wedge\mathbf{V}_k+\overrightarrow{\mathbf{x}_{k2}\mathbf{x}_k}\wedge\mathbf{V}_{k1}+\overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}}\wedge\mathbf{V}_{k2} - \left(\left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k + \overrightarrow{\mathbf{x}_{k2}\mathbf{x}_k} \wedge \mathbf{V}_{k1} + \overrightarrow{\mathbf{x}_k\mathbf{x}_{k1}} \wedge \mathbf{V}_{k2}\right)\cdot\mathbf{n}_j\right)\mathbf{n}_j\Big).$$

So

$$\mathbf{n}'_j[0] = \frac{1}{2\,A_j}\left(\left(\sum_{k\in K_j}\mathbf{e}_{j,k}\wedge\mathbf{V}_k\right) - \left(\left(\sum_{k\in K_j}\mathbf{e}_{j,k}\wedge\mathbf{V}_k\right)\cdot\mathbf{n}_j\right)\mathbf{n}_j\right). \tag{72}$$

Therefore $\mathbf{n}'_j[0]$ is the projection of $\sum_{k\in K_j}\mathbf{e}_{j,k}\wedge\mathbf{V}_k$ on the orthogonal plane to $\mathbf{n}_j$, divided by $2\,A_j$.

In the case where we consider moving only one vertex at once (meaning $\mathbf{V}_{k1}$ & $\mathbf{V}_{k2}$ are null for vertex k), we have :

$$\mathbf{n}'_j[0] = \frac{\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k - \left(\left(\overrightarrow{\mathbf{x}_{k1}\mathbf{x}_{k2}} \wedge \mathbf{V}_k\right)\cdot\mathbf{n}_j\right)\mathbf{n}_j}{2\,A_j} . \tag{73}$$

A.3 Details on the Lambertian Case Using Illumination

All the terms in Equation (62) are explicit at the exception of $\nabla_{\mathbf{x}}R(\mathbf{x},\mathbf{n}_j)$. In fact $\nabla_{\mathbf{x}}R(\mathbf{x},\mathbf{n}_j) = \nabla\rho(\mathbf{x})\mathrm{L}(\mathbf{x},\mathbf{n}_j) + \rho(\mathbf{x})\nabla_{\mathbf{x}}\mathrm{L}(\mathbf{x},\mathbf{n}_j)$ where we denote

$$\mathrm{L}(\mathbf{x},\mathbf{n}_j) = \sum_{l=1}^{n_L} L_l\nu_l(\mathbf{x})(\mathbf{n}_j\cdot\mathbf{l}_l(\mathbf{x})) + E_0 .$$

In the case of a homogeneous albedo (typically in shape from shading) we have $\nabla\rho(\mathbf{x}) = 0$.

According to (54), for $\mathbf{x}$ in $\mathcal{S}_j$, we have $\nabla\rho(\mathbf{x}) = \frac{b\,\nabla a - a\,\nabla b}{b^2}$, where a and b are defined by

$$a = \sum_i I_i(\pi_i(\mathbf{x}))\nu_{\mathcal{S},i}(\mathbf{x}) ,$$

$$b = \sum_i\left(\sum_{l=1}^{n_L^i} L_l^i\nu_{l^i,\mathcal{S}}(\mathbf{x})(\mathbf{n}_j\cdot\mathbf{l}_l^i(\mathbf{x})) + E_0\right)\nu_{\mathcal{S},i}(\mathbf{x}) \tag{74}$$

and where ∇a and ∇b are

$$\nabla a = \sum_i D\pi(\mathbf{x})^T\nabla I_i(\pi_i(\mathbf{x}))\nu_{\mathcal{S},i}(\mathbf{x}) ,$$

$$\nabla b = \sum_i\left(\sum_{l=1}^{n_L^i} L_l^i\nu_{l^i,\mathcal{S}}(\mathbf{x})(\mathbf{n}_j\cdot\nabla\mathbf{l}_l^i(\mathbf{x}))\right)\nu_{\mathcal{S},i}(\mathbf{x}). \tag{75}$$

(We assume here that visibilities are the same for all the points $\mathbf{x}$ on the triangle $\mathcal{S}_j$, or we neglect their variations). For scenes illuminated by far light sources we have $\nabla \mathbf{l}_l^i(\mathbf{x}) \approx 0$, and so $\nabla \rho \approx a'/b$. Moreover, if the light sources are same for all the image, then

$$b = (\sum_i \nu_{\mathcal{S},i}(\mathbf{x})) \left(\sum_{l=1}^{n_L} L_l(\mathbf{n}_j \cdot \mathbf{l}_l) \nu_{l,\mathcal{S}}(\mathbf{x}) + E_0 \right) .$$

Finally, neglecting the variations of the light visibility, we have

$$\nabla_{\mathbf{x}} \mathrm{L}(\mathbf{x}, \mathbf{n}_j) = \sum_{l=1}^{n_L} L_l \nu_l(\mathbf{x}) D\mathbf{l}_l(\mathbf{x})^T \mathbf{n}_j,$$

where $D\mathbf{l}_l(\mathbf{x})^T$ is the transposition of the differential of $\mathbf{l}_l$ (3×3 matrix). In the case of far light sources, we have $\nabla_{\mathbf{x}} \mathrm{L}(\mathbf{x}, \mathbf{n}_j) \approx 0$.

Convex Multi-Region Segmentation on Manifolds

Amaël Delaunoy[1] Ketut Fundana[2] Emmanuel Prados[1] Anders Heyden[3]

[1] INRIA Rhône-Alpes / LJK, Grenoble, France
[2] Applied Mathematics Group, Malmö University, Sweden
and Image Group, Copenhagen University, Denmark
[3] Mathematical Imaging Group, Lund University of Technology, Sweden

Abstract

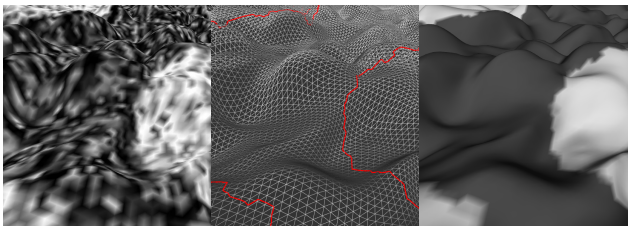
In this paper, we address the problem of segmenting data defined on a manifold into a set of regions with uniform properties. In particular, we propose a numerical method when the manifold is represented by a triangular mesh. Based on recent image segmentation models, our method minimizes a convex energy and then enjoys significant favorable properties: it is robust to initialization and avoid the problem of the existence of local minima present in many variational models. The contributions of this paper are threefold: firstly we adapt the convex image labeling model to manifolds; in particular the total variation formulation. Secondly we show how to implement the proposed method on triangular meshes, and finally we show how to use and combine the method in other computer vision problems, such as 3D reconstruction. We demonstrate the efficiency of our method by testing it on various data.

1. Introduction

Image segmentation aims to partition a given image into several meaningful regions based on certain attributes such as intensity, texture, color, etc. This problem is one of the most challenging and important problems in computer vision. We address the problem of segmenting *data* defined on *manifolds* (typically a 2-surface in $\mathbb{R}^3$) into multiple regions of piecewise constant attributes. The ability to solve such a problem offers significant new possibilities in a number of applications. For example, in 3D reconstruction (see Jin et al. [18]), a segmentation into piecewise constant data of the reconstructed surface allows to naturally introduce constraints on the material of the scene.

1.1. Global Multi-Region Segmentation

Many approaches have been proposed to solve image segmentation problems. In particular, via gradient descents, variational methods for image segmentation have had a great success, such as snakes [20], geodesic active contours [8], geodesic active region [29] and the Chan-Vese models [10]. Yet, the main drawback of those methods is the existence of local minima due to the non-convexity of the energy functionals. Minimizing those functionals by gradient descent methods makes the initialization critical.



(a) (b) (c)

Figure 1: Example of segmentation on a manifold. (a) The Input textured surface. (b) The triangular representation of the surface with the retrieved contour (in red). (c) Surface colored with the mean values of the segmented regions (and surface shading).

To obtain global minima, some previous image segmentation works have used different optimization techniques: For example the graph-cuts in a fully discrete setting, see [6, 21, 22] and the references therein. Nevertheless, while binary segmentation methods based on graph-cuts assure to get a global minima, multi-region segmentation algorithms are based on sequences of graph-cuts which cannot guarantee a global optimization.

Recently, some authors have tried to handle the problem in another direction. Instead of working on the optimization techniques in order to compute the minima of non-convex problems, they have reformulated the energy in order to get a fully convex problem [2, 7, 9, 11, 24, 31, 35]. These segmentation techniques are based on TV-regularizers and aim at finding characteristic functions that minimize the objective functions. Obtaining *global minima* becomes easy and can be done by simply performing a gradient descent. Also

the *initialization problem vanishes*: the algorithm can start from any initialization and obtains the same result. The multi-region segmentation models proposed by [9, 24, 35] are rather similar, the work of Pock et al. [31] (inspired form Ishikawa's [17]) differs from the fact that it deals with ordered labels and uses a regularization term which favors transitions between nearby labels. This makes sense in their stereo application where the ordering is due to depth, but it is not the case when we deal with independent labels. Here, we adopt the model of [9, 24, 35] which is more appropriate to the applications we have in mind (the fact that real-life scenes are made by a finite number of independent materials, in 3D reconstruction problems) and we adapt this *image* labeling model to *manifolds*.

1.2. Data Segmentation on Manifolds

Manifolds such as surfaces are common in computer graphics as well as computer vision. Although data segmentation on surfaces has been recently used on implicit representations (see for example [18, 23]), explicit representations such as triangular meshes are natural and intuitive representations. Mesh representations have been widely used in 3D reconstruction, see for example [1, 12, 33], and recently in [1, 32, 36], which takes advantage of various recent evolution methods. These allow us to naturally deal with topological changes (necessary property e.g. in 3D shape estimation problems formulated within a variational framework). In a number of applications, particularly in graphics, this is the only representation one has at ones disposal. In this paper, after adapting the image segmentation model to manifolds, we show how to implement the proposed method on *triangular meshes*.

Mesh segmentation has been used in computer graphics to decompose meshes into significant parts, but previous work mainly focus on the geometric aspects, the choice and the representations of features to use (like the curvature). We refer to [3] for a recent survey of those techniques, as in this work we focus in segmenting data on the mesh.

However, the problem of segmenting data like texture on manifolds has not received much attention until now, and is quite different from the geometric decomposition of a mesh. In [30], authors take into account both texture information and curvature, but their approach is based on a fast marching algorithm, which needs to be initialized using initial points. Moreover as their segmentation method is not convex, different regions sharing the same properties may result in different labels. The same problem also occurs in [28], where watershed filtering was used. Contrary to those methods, the approach we propose is global and robust to initialization.

Finally, let us note that the segmentation model we consider is based on a total variation regularization. Although this regularization has been previously used on implicit surfaces (see [4] with applications to texture synthesis), it has not received much attention in Lagrangian methods. In the finite elements literature, one can find some papers dealing with the Laplace-Beltrami operator $\nabla_{\mathcal{S}} \cdot \nabla_{\mathcal{S}} u$ which corresponds to the gradient of the squared regularization term $\int_{\mathcal{S}} |\nabla_{\mathcal{S}} u|^2 d\sigma$ (see for example [13, 15]). To our knowledge there does not exist work dealing with the gradient of the total variation in this framework. Let us note here that the Laplace-Beltrami operator is linear while the term associated with the gradient of the total variation $\nabla_{\mathcal{S}} \cdot \frac{\nabla_{\mathcal{S}} u}{|\nabla_{\mathcal{S}} u|}$ is nonlinear. Moreover, the theoretical analysis and the numerical algorithms of [13, 15] are completely based on this linearity property.

1.3. Contributions

First, we adapt the image convex model of [7, 9, 11, 24, 35] to *manifolds*. Then we show how to implement the method when the manifold is represented by a triangular *mesh*. Finally, we explain how our multi-region segmentation method could be incorporated into potential computer vision applications such as *3D reconstruction*.

2. Multi-Region Segmentation Model

In this section, we describe the convex image segmentation model we propose. To make this model comprehensible and intuitive, let us first remind of the region-based active contour model of Chan and Vese [10]. Here we show that the energy functional of Chan and Vese, which is the piecewise constant case of the Mumford-Shah model ([27]), can be recast as a convex functional in order to find the global minimizer of the original energy functional.

2.1. Convex Two-Phases Model

The Chan-Vese model [10], which is formulated in the level set framework, partition a given image into two subregions. For a given image I, the idea is to find a subset Σ of a bounded domain $\Omega \subset \mathbb{R}^N$, whose boundary $\partial\Sigma$ is represented by the zero level set of function $\phi : \Omega \to \mathbb{R}^N$. This is done by minimizing the energy functional

$$\min_{\phi, c_1, c_2} \int_\Omega \Big\{ H_\epsilon(\phi)(I(\mathbf{w}) - c_1)^2 + (1 - H_\epsilon(\phi))(I(\mathbf{w}) - c_2)^2 + \lambda \left|\nabla H_\epsilon(\phi)\right| \Big\} \, d\omega \,, \quad (1)$$

where $\lambda \in \mathbb{R}$, $c_1, c_2 \in \mathbb{R}$ and H_ϵ is a regularized Heaviside function, which models a characteristic function (see [10]).

Since the energy functional (1) is not convex, minimizing it by gradient descent methods can get stuck in local minima. By relaxing the characteristic function $H_\epsilon(\phi)$ by an arbitrary function u bounded between 0 and 1, Chan et

al. [11] showed that minimizing (1) can be rewritten as the following convex minimization problem:

$$\min_{0\leq u\leq 1}\Big\{\int_\Omega \Big\{u(\mathbf{w})(I(\mathbf{w})-c_1)^2 + (1-u(\mathbf{w}))(I(\mathbf{w})-c_2)^2\Big\}\,d\omega + \lambda\int_\Omega |\nabla u|\,d\omega\Big\}, \quad (2)$$

c_1 and c_2 being fixed, in $\mathbb{R}$. As proved in [7, 11], if $u(x)$ is a minimizer of (2), then for a.e. $\mu \in [0,1]$, the set $\Sigma(\mu) = \{x \in \Omega, u(x) > \mu\}$ is a minimizer of the Mumford-Shah functional [27], implying that *the solution to (1) can be obtained by thresholding* u *at any arbitrary threshold between* 0 *and* 1.

2.2. Extension to Multi-Region Segmentation

Recently, several authors [9, 24, 35] have extended the convex formulation (2) to multi-region segmentation:

$$\min_{\mathbf{u}\in K}\Big\{\int_\Omega <\mathbf{u}(\mathbf{w}),\mathbf{s}(\mathbf{w})> + \lambda\,|\nabla\mathbf{u}(\mathbf{w})|\,d\omega\Big\}, \quad (3)$$

where K is the set of function $\mathbf{u}:\Omega\to\mathbb{R}^m$ such that for all $\mathbf{w}\in\Omega$ and $p\in[1..m]$, $\mathbf{u}_p(\mathbf{w})\geq 0$ and $\sum_{p=1}^m \mathbf{u}_p(\mathbf{w}) = 1$. $|\nabla\mathbf{u}(\mathbf{w})|$ corresponds to $\sqrt{\sum_p |\nabla\mathbf{u}_p(\mathbf{w})|^2}$, where $|.|$ denotes the L^2 norm. m denotes the number of labels and $\mathbf{s}(\mathbf{w})$ is an m-dimensional vector; $s_p(\mathbf{w})$ indicates the affinity of the data at point $\mathbf{w}$ with class p. The convex domain naturally allow direct competition between the labeling.

3. Multi-Region Segmentation on Manifolds

In this section we extend the multi-region convex model (3) on a manifold, and we show how to optimize the associated energy for a manifold represented by a mesh. To our best knowledge, these convex formulations (2,3) have been defined only on open subsets of $\mathbb{R}^N$ which correspond to image domains, as described in the previous section.

Let $\mathcal{S}$ be a Riemannian manifold. Typically, $\mathcal{S}$ could be a smooth 2D surface of $\mathbb{R}^3$. Energy (3) is adapted as follows:

$$\min_{\mathbf{u}\in K}\Big\{\int_{\mathcal{S}} <\mathbf{u}(\mathbf{x}),s(\mathbf{x})> + \lambda|\nabla_{\mathcal{S}}\mathbf{u}(\mathbf{x})|\,d\sigma\Big\}, \quad (4)$$

where now the functions $\mathbf{u}$ are defined on $\mathcal{S}$ instead of Ω, $|.|$ is the Riemannian norm, $\nabla_{\mathcal{S}}$ is the intrinsic gradient on $\mathcal{S}$ and $d\sigma$ is the manifold's element measure (surface's area measure for 2D manifolds).

Now, let us consider a manifold represented by a mesh. The following results apply to manifolds with any topology. Let $\mathbf{X}$ be a (piecewise linear) polyhedron representation of the surface $\mathcal{S}$, defined by a set of vertices $\mathbf{x}_k : \mathbf{X} = \{\mathbf{x}_k\}$ and let l be the cardinality of $\mathbf{X}$ (the number of vertices). As in the finite elements literature, we define $\phi_k : \mathcal{S}\to\mathbb{R}$ as the piecewise affine, interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i\neq k$. The vector valued field $\mathbf{U} = \{\mathbf{u}_k\}$ is defined on all vertices $\mathbf{x}$ of the polyhedron $\mathbf{X}$. $\mathbf{U}$ can be naturally extended on $\mathcal{S}$ by a piecewise affine vector valued field on $\mathcal{S}$. We denote this extension $\mathbf{u}(\mathbf{x}) = \sum_k \mathbf{u}_k\phi_k(\mathbf{x})$. To make the paper easier to read and because of space limitations, we assume that the manifold is a 2D surface of $\mathbb{R}^3$. However, the following method applies to any dimension. Let $\mathcal{S}_j$ be the jth triangle of the mesh. The multi-region segmentation energy can then be rewritten as

$$\sum_j\sum_k \Big\langle \mathbf{u}_k, \int_{\mathcal{S}_j}\phi_k(\mathbf{x})s(\mathbf{x})d\sigma\Big\rangle + \lambda\int_{\mathcal{S}_j}|\nabla_{\mathcal{S}}\mathbf{u}(\mathbf{x})|d\sigma, \quad (5)$$

where $\mathbf{u}$ is constrained to be in K. The first term of (5) is explicitly written with respect to $\mathbf{U}$. In order to make the total variation term explicit with respect to $\mathbf{U}$, we first consider a local paramtrization (α,β) on the manifold. Following [14, 19], we rewrite the right term of Equation (5) using fundamental forms:

$$\nabla_{\mathcal{S}}\mathbf{u} = \begin{bmatrix}\frac{\partial\mathbf{x}}{\partial\alpha} & \frac{\partial\mathbf{x}}{\partial\beta}\end{bmatrix}\begin{bmatrix}E & F\\ F & G\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{u}_\alpha\\ \mathbf{u}_\beta\end{bmatrix}\text{, and then}$$

$$|\nabla_{\mathcal{S}}\mathbf{u}| = \sqrt{\begin{bmatrix}\mathbf{u}_\alpha & \mathbf{u}_\beta\end{bmatrix}\begin{bmatrix}E & F\\ F & G\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{u}_\alpha\\ \mathbf{u}_\beta\end{bmatrix}},$$

where $\mathsf{E} = \frac{\partial\mathbf{x}}{\partial\alpha}\cdot\frac{\partial\mathbf{x}}{\partial\alpha}$, $\mathsf{F} = \frac{\partial\mathbf{x}}{\partial\alpha}\cdot\frac{\partial\mathbf{x}}{\partial\beta}$ and $\mathsf{G} = \frac{\partial\mathbf{x}}{\partial\beta}\cdot\frac{\partial\mathbf{x}}{\partial\beta}$ are coefficients of the first fundamental form (see [14, 19]). $\mathbf{u}_\alpha$ and $\mathbf{u}_\beta$ are partial derivatives of $\mathbf{u}$ with respects to α and β respectively. Considering the mesh representation, we parametrize the triangle $\mathcal{S}_j$ by $\mathbf{x}(\alpha,\beta) = \mathbf{x}_{j,1} + \alpha\,\overrightarrow{\mathbf{x}_{j,1}\mathbf{x}_{j,2}} + \beta\,\overrightarrow{\mathbf{x}_{j,1}\mathbf{x}_{j,3}}$ where $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$ and $\mathbf{x}_{j,3}$ are the three vertices associated with the triangle $\mathcal{S}_j$ and where $(\alpha,\beta)\in T = \{(\alpha,\beta)|\alpha\in[0,1]\text{ and }\beta\in[0,1-\alpha]\}$. We then have $\int_{\mathcal{S}_j}|\nabla_{\mathcal{S}}\mathbf{u}(\mathbf{x})|d\sigma =$

$$\int_T\sqrt{\sum_p \mathbf{u}_{p_\alpha}^2\mathsf{G} - 2\mathbf{u}_{p_\alpha}\cdot\mathbf{u}_{p_\beta}\mathsf{F} + \mathbf{u}_{p_\beta}^2\mathsf{E}}\;d\alpha d\beta\,. \quad (6)$$

$\mathbf{u}_{p_\alpha}$ and $\mathbf{u}_{p_\beta}$ are partial derivatives of $\mathbf{u}_p$ with respects to α and β respectively. Here the reader will easily verify that E, F, G, $\mathbf{u}_{p_\alpha}$ and $\mathbf{u}_{p_\beta}$ are constant functions on $\mathcal{S}_j$ and that their respective values are equal to $\mathsf{E}_j = |\mathbf{x}_{j,2}-\mathbf{x}_{j,1}|^2$, $\mathsf{F}_j =< \mathbf{x}_{j,2}-\mathbf{x}_{j,1},\mathbf{x}_{j,3}-\mathbf{x}_{j,1} >$, $\mathsf{G}_j = |\mathbf{x}_{j,3}-\mathbf{x}_{j,1}|^2$, $\mathbf{u}_{p_\alpha}^j = \mathbf{u}_{j,2_p} - \mathbf{u}_{j,1_p}$ and $\mathbf{u}_{p_\beta}^j = \mathbf{u}_{j,3_p} - \mathbf{u}_{j,1_p}$, where $\mathbf{u}_{j,1}$, $\mathbf{u}_{j,2}$ and $\mathbf{u}_{j,3}$ are the values of $\mathbf{u}$ at vertices $\mathbf{x}_{j,1}$, $\mathbf{x}_{j,2}$ and $\mathbf{x}_{j,3}$ respectively. Now the term inside the integral of (6) does not depend on α and β. The convex multi-region

segmentation energy on the meshed manifold becomes:

$$E(\mathbf{U}) = \sum_j \sum_k \left\langle \mathbf{u}_k, \int_{\mathcal{S}_j} \phi_k(\mathbf{x}) s(\mathbf{x}) d\sigma \right\rangle + \frac{\lambda}{2} \sum_j \sqrt{\sum_p {\mathbf{u}_{p_\alpha}^j}^2 \mathsf{G_j} - 2\mathbf{u}_{p_\alpha}^j \cdot \mathbf{u}_{p_\beta}^j \mathsf{F_j} + {\mathbf{u}_{p_\beta}^j}^2 \mathsf{E_j}} \,. \quad (7)$$

3.1. Optimization Method

When the (surface) manifold is represented by a mesh, the convex multi-region segmentation model then leads to optimizing the convex energy (7) with respect to $\mathbf{U} \in \mathbb{R}^{l \times m}$, with the convex constraint $\mathbf{U} \in K$; K being the set $\{\mathbf{U} \text{ s.t. } \forall k,\ \sum_p \mathbf{u}_{kp} = 1 \text{ and } \forall p,\ \mathbf{u}_{kp} \geq 0\}$. This convex constrained optimization problem on $\mathbb{R}^{l \times m}$ can be solved by the projected gradient method [5], which consists in generating the sequence U^t via

$$\mathbf{U}^{t+1} = Proj_K(\mathbf{U}^t - \tau \nabla E(\mathbf{U}^t)) \,, \quad (8)$$

for a fixed time step $\tau > 0$, until $|\mathbf{U}^t - \mathbf{U}^{t-1}|_\infty \leq \delta$, a small constant. $Proj_K$ is the projection on the convex set K. In other words, we iteratively process gradient descent steps and projections of the $\mathbf{u}_k$ on the set K. These projections can be done via Michelot's algorithm [26]. From energy (7) we easily obtain

$$\frac{\partial E}{\partial \mathbf{u}_{kp}}(\mathbf{U}) = \sum_{j \in N(k)} \left[\int_{\mathcal{S}_j} \phi_k(\mathbf{x}) s(\mathbf{x}) d\sigma \right]_p - \frac{\lambda}{2}\, Q\, (\xi + \epsilon)^{-\frac{1}{2}}, \quad (9)$$

where $Q = (\mathbf{u}_{j,2_p} - \mathbf{u}_{kp})(\mathsf{G}_j - \mathsf{F}_j) + (\mathbf{u}_{j,3_p} - \mathbf{u}_{kp})(\mathsf{E}_j - \mathsf{F}_j)$, ξ is the term in the squared root of (7), and $N(k)$ is the 1-ring neighborhood of vertex k. As in [11], we regularize the term ξ by incorporating a small value ϵ inside the squared root to avoid instabilities when the gradient of $\mathbf{u}$ is 0.

Let us remind now that, as underlined by [16], the notion of gradient depends on the underlying scalar product. If we chose the pointwise scalar product $< \mathbf{U}, \mathbf{V} >_{\mathsf{pw}} = \sum_k < \mathbf{u}_k, \mathbf{v}_k >$, then the components of $\nabla E(\mathbf{U}^t)$ directly coincide with $\frac{\partial E}{\partial \mathbf{u}_{kp}}(\mathbf{U})$. Nevertheless the associate pointwise metric is not efficient for minimizing energies of the form $\int_{\mathcal{S}} f(\mathbf{u}(\mathbf{x}))\, d\sigma$ since the distance between two discrete fields $\mathbf{U}$ and $\mathbf{V}$ does not take into account the area of the triangle. On the other hand, the L^2 scalar product $< \mathbf{U}, \mathbf{V} >_{L^2} = \int_{\mathcal{S}} < \mathbf{u}(\mathbf{x}), \mathbf{v}(\mathbf{x}) > d\sigma$ is much more relevant. Also in this case the gradient becomes

$$\nabla E(\mathbf{U}) = M^{-1} \frac{\partial E}{\partial \mathbf{U}}(\mathbf{U}) \,, \quad (10)$$

where the matrix M is the mass matrix defined by $M_{ij} = Id_m \int_{\mathcal{S}} \phi_i(x) \phi_j(x)\, d\sigma$. Moreover one classically approximates M by the diagonal mass lumping $\tilde{M}$, where $\tilde{M}_{ii}$ is the area of the Voronoi dual cell of $\mathbf{x}_i$ times the identity matrix Id_m, see e.g. [16].

3.2. Applications

In the previous sections the data term of the segmentation model s is assumed to be known (Equation 4). In the applications, this term also depends on some parameters that have to be optimized. The convex problem can be solved by alternating optimization of the parameters in a bi-convex way. For fixed parameters of s we update $\mathbf{u}$ and vice-versa. $\mathbf{u}$ is updated according to the method presented in section 3.1. In practice we update the parameters of s every r update iterations of $\mathbf{u}$ (r is chosen arbitrary; we fix $\mathsf{r} = 10$ in our experiments).

Piecewise Constant Data Segmentation

Let us consider the case where the data we want to segment are assumed to be piecewise constant. Here a natural expression for $s_p(\mathbf{x})$ is to use the squared error between the scalar or vector-valued data $C(\mathbf{x})$ at the point $\mathbf{x}$ and the value μ_p associated with the label p (μ_p having the same dimension as the data):

$$s_p(\mathbf{x}) = (C(\mathbf{x}) - \mu_p)^T (C(\mathbf{x}) - \mu_p) \,.$$

The optimization of the energy (4) with respect to μ_p gives:

$$\mu_p = \frac{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x})\, C(\mathbf{x}) d\sigma}{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x}) d\sigma} \,,$$

which corresponds to the mean value of the data of the associated region. Note that the previous model can be easily extended to any probablility density function D_p. For example, D_p can be a multivariate gaussian density function of mean μ_p and covariance Σ_p, and then we would have:

$$s_p(\mathbf{x}) = -\ln(D_p(\mathbf{x}, \mu_p, \Sigma_p)) \text{ , with}$$

$$D_p(\mathbf{x}, \mu_p, \Sigma_p) = \frac{1}{\sqrt[m]{2\pi}|\Sigma|} e^{-\frac{1}{2}(C(\mathbf{x}) - \mu_p)^T \Sigma_p^{-1} (C(\mathbf{x}) - \mu_p)} \,.$$

Segmentation in 3D Reconstruction Problems

Such segmentation framework can be incorporated in 3D Reconstruction applications. In such applications, it can be interesting to segment a particular region, or all parts of the surface sharing the same reflectance properties. In 3D reconstruction, most of the variational methods yield to minimizing an energy of the form

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{S}} g(\mathbf{x}) \frac{\mathbf{x}_i \cdot \mathbf{n}}{\mathbf{x}_{i,z}^3} \nu_{\mathcal{S}}(\mathbf{x})\, d\sigma \,, \quad (11)$$

see for example [12, 18, 19]. Moreover, if we choose

$$g(\mathbf{x}) = \sum_{p=1}^{m} \mathbf{u}_p(\mathbf{x}) (I_i(\pi_i(\mathbf{x})) - \mu_p)^T (I_i(\pi_i(\mathbf{x})) - \mu_p) \,,$$

where $\pi_i(\mathbf{x})$ is the projection of the surface point $\mathbf{x}$ into the ith image and $I_i : w \mapsto I_i(w)$ is the function which associates to each pixel w, its color on the ith image. We then get an extension of the stereoscopic segmentation method proposed by [34] to the case where the surface is composed of more than two regions of piecewise constant radiance. Also, contrary to our method, the segmentation approach proposed in [34] is subject to local minima. Finally, the optimization of the energy (4) with respect to μ_p gives:

$$\mu_p = \frac{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x}) \ \sum_i I_i(\pi_i(\mathbf{x})) \frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) d\sigma}{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x}) \sum_i \frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) d\sigma} .$$

If we chose $g(\mathbf{x}) =$

$$\sum_{p=1}^{m} \mathbf{u}_p(\mathbf{x})(I_i(\pi_i(\mathbf{x})) - \rho_p \mathbf{N}(\mathbf{x})\cdot\mathbf{L})^T (I_i(\pi_i(\mathbf{x})) - \rho_p \mathbf{N}(\mathbf{x})\cdot\mathbf{L}),$$

where $\mathbf{N}(\mathbf{x})$ is the normal to the surface at the point $\mathbf{x}$ and $\mathbf{L}$ is the vector corresponding to the light source illuminating the scene, then we get an extension of the (Lambertian) multi-view shape from shading method proposed by [18] for surfaces with piecewise constant albedo. In the same way, contrary to our approach, the method proposed by [18] is limited to two regions segmentation and is strongly subject to local minima. The optimization of energy (4) with respect to the albedo gives:

$$\rho_p = \frac{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x}) \ \sum_i I_i(\pi_i(\mathbf{x})) \mathbf{N}(\mathbf{x})\cdot\mathbf{L} \frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) d\sigma}{\int_{\mathcal{S}} \mathbf{u}_p(\mathbf{x}) (\mathbf{N}(\mathbf{x})\cdot\mathbf{L})^2 \sum_i \frac{\mathbf{x}\cdot\mathbf{n}}{\mathbf{x}_z^3} \nu_{\mathcal{S},i}(\mathbf{x}) d\sigma} .$$

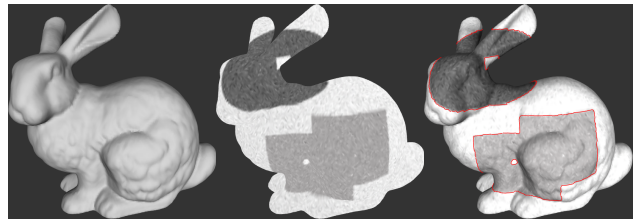
The theoretical and experimental study of these algorithms will be the topic a forthcoming paper.

4. Experiments

In order to validate the proposed multi-region segmentation approach on meshes, we present different experiments on synthetic as well as realistic data. In practice as explained in previous section, the segmentation is solved by alternating between region parameters and the segmentation variable $\mathbf{U}$, with a known number of regions. The algorithm complexity is linearly dependent on the number of facets and the number of classes. Experiments have been runned on a 2.66GHz linux machine and take about 200 seconds on a mesh of 200,000 facets and for a 4 regions segmentation. The values of λ have been manually chosen in each example but a value of 0.01 gives reasonnable results in most cases.

4.1. The Two Region Case

Figures 2 and 3 show examples of our algorithm using a synthetic image mapped onto a mesh for the Stanford bunny model. Noise has been added to the image. Here, we show that our algorithm performs well on the given example and that the final solution is binary. Moreover it is robust to the initialization of the scalar function $\mathbf{U}$. Note that the retrieved solution that has been displayed is the auxiliary value $\mathbf{u}$, and not the segmented constant values μ_1 and μ_2. Also because the energy functional is convex in $\mathbf{u}$ only and the values μ_1 and μ_2 are optimized during the evolution, they can be assigned to the region corresponding to either $\mathbf{u} = (1, 0)$ or $\mathbf{u} = (0, 1)$, this explains why the last initialization do not show the same values of $\mathbf{u}$ but an inverted one. In practice, although the total functional is not fully convex, we obtain the same results and really similar μ_1 and μ_2 for each example. We respectively obtain ($\mu_1 = 140.778$, $\mu_2 = 231.003$), ($\mu_1 = 140.746$, $\mu_2 = 231.01$), ($\mu_1 = 140.75$, $\mu_2 = 231.03$) and ($\mu_1 = 230.992$, $\mu_2 = 140.765$) for the four different initializations. Note that in the last column, values of μ_1



(a) (b) (c)

Figure 2: Segmentation result on the synthesized Stanford bunny surface. (a) Input shaded object. (b) Input mesh with synthetic texture mapping. (c) Input textured mesh (shaded visualization) and final contour (in red).

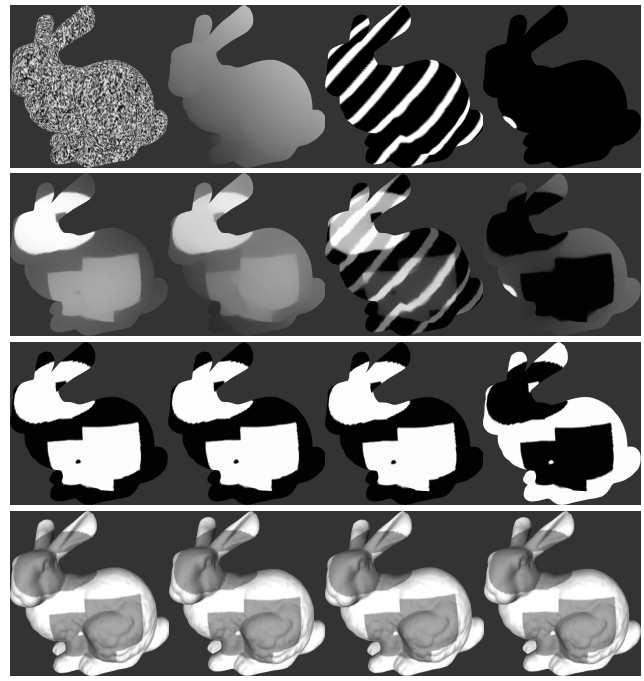


Figure 3: The evolution on the synthesized bunny surface. Different initialization of $\mathbf{U}$ (first row); Intermediate values of $\mathbf{U}$ (second row); The obtained solution $\mathbf{U}$ (third row); The obtained mean values (fourth row) with shading.

and μ_2 are inverted and the solution $\mathbf{u}$ as well. In this example, geodesic active contours or level sets methods would tend to the closest local minima from the initialization as the texture is not clearly binary. Nevertheless, as the method is global here, segmenting a particular region should be done using additional cues.

Figure 4 present segmented surfaces from real-world textures that have been mapped onto a mesh, in the case of the two-phases segmentation. We show different examples from classical images used in segmentation. Note that the segmentation is done on the mesh using the method described in this paper and not on an image. The experiments show three different non binary images and their segmentation into two different regions. As expected the results are binary even though the initial values of the segments are random values. The mean values of each region is estimated during the process as described before, and the parameters λ can be adjusted to add more smoothness to the segmentation. As shown by experiments, even though the initialization is random and the parameters of each region are computed during the evolution, the algorithm still converges to the desired solution as a binary solution.

Figure 4: Segmentation results on meshes in the two-region segmentation on three different examples. From top to bottom: Input textured mesh; Mesh shape where the segmentation is performed and the initial random value of one component of $\mathbf{U}$; Recovered mean values of each region; Segmented object.

4.2. Dealing with Multiple Regions

Here we show the efficiency of the proposed method when dealing with multiple regions. Different examples are shown, first with synthetic textures on which noise has been added, and then on meshes textured by real-world images like previous examples. Note that the number of regions is initially given and is not automatically estimated.

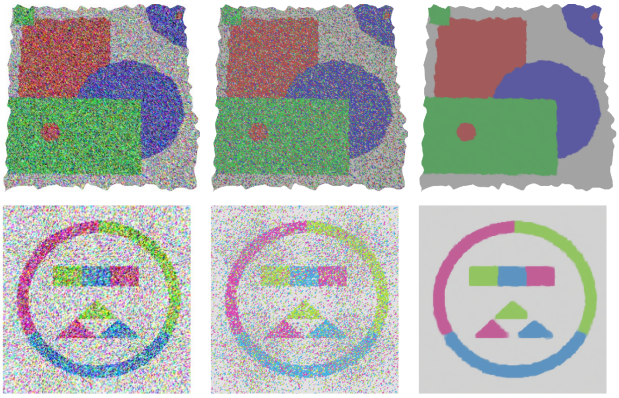


Figure 5: Segmentation results on meshes in the multi-region case on synthesized examples inspired from [24]. (a) Input textured mesh (same shape as previous Butterfly and Horses data). (b) Clustering using K-Means algorithm. (c) Recovered mean values of each region obtained by our approach.

In Figure 5, the experiment shows noisy texture on meshes, the segmentation result using K-Means, and the result of our TV-based algorithm on meshes. Because the K-Means algorithm does not take into account the spatial coherence of points, the result is noisy. On the other hand, the TV regularization allows coherence in the scene and the segmentation is close to the expected solution. In addition to be robust to initialization, our approach is robust to noise.

We then tested our multi-region segmentation approach on various data from real-world images [25], see Figure 6. Let us emphasize here that the initialization was random and the number of regions was initially given.

As an example, we applied our approach to segment mean curvature on a mesh using three different regions. Figure 7 shows that we are able to segment some concave and convex parts of the mesh.

Finally, in Figure 8, we show the examples of a 3D mesh obtained by 3D reconstruction algorithms, as the one in [33, 36]. The last row shows the obtained color-based labeling (into three regions). Even though the texture is far from being binary, the segmentation is the expected one. For instance in the result, we nicely recover the skin, the pant and the shirt. Here again, initialization was random.

For comparison of the convex *image* multi-region segmentation model (3) with other methods, we refer to [24] which shows quantitative and qualitative comparisons with belief propagation, sequential belief propagation, graph cuts

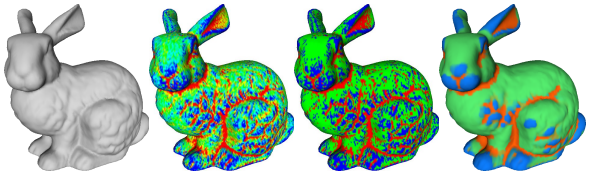
Figure 8: Segmentation result on a colored mesh obtained by multiview stereo algorithm. Front view (top row) and back view (bottom row). Original input colored mesh and the associated 3D shape (left). Result of the segmentation into three regions obtained by our algorithm (middle). Recovered mean values displayed for each region (right).

Figure 6: Segmentation results on meshes in the multi-region case. Top row: Horse data set and its segmentation for three regions. Bottom row: Four regions labeling of the Butterfly data. (a) Input textured mesh; (b) Recovered mean values of each region obtained by our approach; (c) One of the segmented regions.

with alpha-expansion, graph cuts with alpha-beta swap and sequential tree reweighted belief propagation methods. The experiments show that the generated labeling is comparable to state-of-the-art discrete optimization methods.

5. Conclusions

In this paper we propose a variational method for segmenting *data* on *manifolds* into regions of constant properties. The *convex* formulation makes the proposed model



(a) (b) (c) (d)

Figure 7: Segmentation results on mesh curvature in 3 regions. (a) Input mesh. (b) Mean Curvature visualization. (c) Simple thresholding of the mean curvature. (d) Segmentation result of the mean curvature into three regions with our approach.

robust to initialization. Moreover, the total variation regularizer makes the method *robust to noise*. We show how to implement the method, in particular how to compute the gradient of the total variation term, when the surface has a discrete representation as triangular meshes. We have demonstrated the efficiency of our method by testing it on various synthetic and realistic data from computer vision applications.

Acknowledgements

Amaël Delaunoy and Emmanuel Prados were supported by the Flamenco project ANR-06-MDCA-007, and Ketut Fundana and Anders Heyden were funded by the VISION-TRAIN RTN CT-2004-005439 Marie Curie Action within the EC's Sixth Framework Programme. We thank Andrei Zaharescu and Adrian Hilton for sharing the dancer data with the Perception group.

References

[1] E. Aganj, J.-P. Pons, F. Sgonne, and R. Keriven. Spatio-temporal shape from silhouette using four-dimensional delaunay meshing. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct 2007.

[2] B. Appleton and H. Talbot. Globally minimal surfaces by continuous maximal flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1), January 2006.

[3] M. Attene, S. Katz, M. Mortara, G. Patanè, M. Spagnuolo, and A. Tal. Mesh segmentation - a comparative study. 2006.

[4] M. Bertalmio, G. Sapiro, L.-T. Cheng, and S. Osher. Variational problems and pdes on implicit surfaces. *Variational and Level Set Methods in Computer Vision*, 2001.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2003.

[6] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *IEEE International Conference on Computer Vision*, 2001.

[7] X. Bresson, S. Esedoglu, P. Vandergheynst, J. P. Thiran, and S. J. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, June 2007.

[8] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, (1), 1997.

[9] A. Chambolle, D. Cremers, and T. Pock. A convex approach for computing minimal partitions. Technical Report R.I.649, CMAP CNRS, November 2008.

[10] T. Chan and L. Vese. Active contour without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[11] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *Journal of Applied Mathematics*, 2006.

[12] A. Delaunoy, E. Prados, P. Gargallo, J.-P. Pons, and P. Sturm. Minimizing the multi-view stereo reprojection error for triangular surface meshes. In *British Machine and Vision Conference, Leeds, UK*, 2008.

[13] A. Demlow and G. Dziuk. An adaptive finite element method for the laplace-beltrami operator on implicitly defined surfaces. *SIAM Journal of Numerical Analysis*, (1), 2007.

[14] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, February 1976.

[15] G. Dziuk. Finite elements for the beltrami operator on arbitrary surfaces. In S. B. . Heidelberg, editor, *Partial Differential Equations and Calculus of Variations*, pages 142–155. Lecture Notes in Mathematics, 2006.

[16] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, and M. Desbrun. Generalized surface flows for mesh processing. In *Eurographics Symposium on Geometry Processing*, 2007.

[17] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25, 2003.

[18] H. Jin, D. Cremers, D. Wang, E. Prados, A. Yezzi, and S. Soatto. 3-d reconstruction of shaded objects from multiple images under unknown illumination. *International Journal of Computer Vision*, 76(3), March 2008.

[19] H. Jin, A. Yezzi, Y.-H. Tsai, L.-T. Cheng, and S. Soatto. Estimation of 3d surface shape and smooth radiance from 2d images: A level set approach. *J. Sci. Comput.*, (1-3), 2003.

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[21] P. Kohli and P. H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Dec. 2007.

[22] Kolmogorov and Zabih. What energy functions can be minimized via graph cuts. *IEEETPAMI: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[23] M. Krueger, P. Delmas, and G. Gimel'Farb. Active contour based segmentation of 3d surfaces. In *European Conference on Computer Vision*, 2008.

[24] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnoerr. Convex multi-class image labeling by simplex-constrained total variation. Technical report, University of Heidelberg, oct 2008.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.

[26] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of rn. *Journal of Optimization Theory and Applications*, 50(1):195–200, 1986.

[27] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Comm. on Pure and Applied Math.*, XLII(5):577–685, 1988.

[28] D. L. Page, A. F. Koschan, and M. A. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. *Computer Vision and Pattern Recognition*, 2003.

[29] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97:259–282, 2005.

[30] G. Peyre and L. Cohen. Surface segmentation using geodesic centroidal tesselation. *International Symposium on 3D Data Processing, Visualization and Transmission*, Sept. 2004.

[31] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multi-label problems. In *European Conference on Computer Vision*, pages III: 792–805, 2008.

[32] J.-P. Pons and J.-D. Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, Jun 2007.

[33] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 2007.

[34] A. Yezzi and S. Soatto. Stereoscopic segmentation. *IEEE International Conference on Computer Vision*, 2001.

[35] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Proceedings of Vision, Modeling, and Visualization*, 2008.

[36] A. Zaharescu, E. Boyer, and R. Horaud. Transformesh: a topology-adaptive mesh-based approach to surface evolution. In *Proceedings Asian Conference on Computer Vision*, Tokyo, Japan, November 2007.

Towards Full 3D Helmholtz Stereovision Algorithms

Amaël Delaunoy[1], Emmanuel Prados[1], and Peter N. Belhumeur[2]

[1] INRIA Grenoble / LJK
[2] Columbia University

Abstract. Helmholtz stereovision methods are limited to binocular stereovision or depth maps reconstruction. In this paper, we extend these methods to recover the full 3D shape of the objects of a scene from multiview Helmholtz stereopsis. Thus, we are able to reconstruct the complete three-dimensional shape of objects made of any arbitrary and unknown bidirectional reflectance distribution function. Unlike previous methods, this can be achieved using a full surface representation model. In particular occlusions (self occlusions as well as cast shadows) are easier to handle in the surface optimization process. More precisely, we use a triangular mesh representation which allows to naturally specify relationships between the geometry of a point of the scene and its surface normal. We show how to implement the presented approach using a coherent gradient descent flow. Results and benefits are illustrated on various examples.

1 Introduction

Reconstructing shape and appearance of objects from images is still one of the major problems in computer vision and graphics. In this work, we are interested in recovering a full and dense representation of the object's three-dimensional shape. Multiview reconstruction systems are commonly used to estimate such a model, as they provide information from many viewpoints around the object of interest. Among these approaches, variational methods have been popular because they can be used to solve a wide variety of vision problems. The idea is to minimize an energy functional that depends on the considered object surface and on the input images, whose minima is reached at the object of interest. Many methods have been proposed in order to solve this problem, but these approaches are often limited in the kind of appearance they can handle. To overcome these limitations, we exploit Helmholtz reciprocity and propose a single framework for normal estimation and normal integration using a triangular mesh-based deformable model.

1.1 Helmholtz Stereovision: a Reconstruction Approach for Real World Objects

Most of the multiview reconstruction algorithms rely on image correspondences (as done for instance in multiview stereo [1]) or shading (using the normal information in multiview shape from shading [2, 3] or multiview photometric stereo [4]). When texture information (stereo case) is good enough or Lambertian assumption is sufficiently verified, those methods have been proved to give good results with surfaces that are nearly. They then obtain either accurate correspondences or accurate normal estimates.

But when the scene is not Lambertian, which is the case for most (if not all) real world scenes, such cues are not valid and algorithms fail to reconstruct accurately the surface. In order to solve this problem, many alternatives have been proposed. Some authors consider specular highlights as outliers [4], and consider a large number of images in order to compensate. Some others modify the input images to have specular free images, and be photometric invariant [5, 6]. Another approach is to have a robust similarity measurement [7, 8]. All these methods try to compensate the non-Lambertian components in order to run reconstruction algorithms designed for the Lambertian case.

Some authors consider another strategy: they propose to use more general reflectance / radiance parametric models. See for instance [9–11]. In practice, these approaches suffer from several limitations. First of all the reflectance model has to be known in advance and this constrains scene to be composed by materials consistent with the chosen reflectance model. Such algorithms tend to solve non-linear systems of thousands of variables (one reflectance / radiance model per surface point), or need additional assumptions (single or fixed number of materials, single specular component, etc.). Those models are difficult to be optimized and generally require to alternatively estimate the reflectance and the shape. They are numerically unstable and easily tend to get stuck in local minima [3]. Moreover, the algorithms are generally ill-posed, and then require strong regularization which over-smooth the obtained results. Finally, the reflectance and illumination models need also to be approximated. Although such algorithms show reasonable results for perfect synthesized scenes, their application to real-world scenes is complex and requires accurate camera and light calibration.

A different approach introduced by [12] uses radiance samples of reflectance exemplars to match it with the observed images. A direct matching allows to estimate the surface normal in order to reconstruct the shape by normal integration. Although it can deal with many materials and anisotropic BRDF, material samples are needed. This can be a restrictive assumption in concerned applications.

During the last decade, some authors have proposed to use Helmholtz reciprocity in order to perform 3D reconstruction [13–17]. In practice, Helmholtz reciprocity is exploited by taking a pair of images under a single light source, where camera centers and light positions are exchanged at each shot. It uses the fact that in this particular setup, for a single reciprocal pair, the relationship between two radiances of a single surface point is independent of the reflectance. Contrary to works described in previous paragraphs, methods based on Helmholtz reciprocity [13–17] allow to accurately estimate the normals at one point, independently of the reflectance model. This can be used to obtain a 3D surface. In this context, modeling the reflectance, having material samples or being photometric invariant is not required. Nevertheless, contrary to most of multiview stereovision algorithms, the state of the art in Helmholtz reconstruction is limited to depth map reconstructions. In this paper, we push the envelope by proposing a Full 3D multiview Helmholtz stereovision method.

1.2 A Surface-Based Approach

Until now, all the previous Helmholtz reconstruction methods were camera-view centered. On the contrary, we propose to change the surface representation and to adopt an

object-centered strategy. Thus, instead of using a 2.5D surface as it used to be done previously, we represent the object's surface by a closed and dense 2D manifold embedded in the euclidean 3D space. The interest of this choice is two-fold. First this allows to naturally recover a full 3D surface when the other approaches only recover a depth map (or a needle map). Secondly, this allow to easily and properly handle visibility (shadows as well as self occlusions). Figure 1 sheds light on these advantages: the picture on the left illustrates the case of conventional Helmholtz Stereovision methods, that recover a surface based on a virtual view (here in red). The drawing on the right illustrates the proposed approach which is surface-based instead of view-based reconstruction. The red contours show the optimal surface each method can recover. Clearly, the conventional approaches are confronted with difficulties with visibility since it generates discontinuities in the depth maps and the needle maps. Also, this difficulty spreads at the integration steps which behave badly in presence of discontinuities.

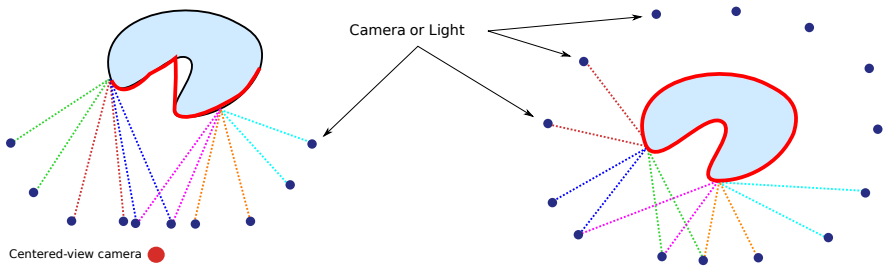


Fig. 1. Recovered surfaces by conventional Helmholtz Stereopsis versus proposed approach (surface in red). Left: case of conventional Helmholtz Stereo methods which recovers the shape based on a virtual view (in red). Right: the proposed surface-based approach.

In this work, as we will discuss in section 3, we represent the 2D surface using a triangle mesh. In our context, where the surface normals play a key role, this representation offers significant advantages. In particular, it intrinsically links a depth to its surface normal which is, in the case of piecewise planar surfaces like triangular meshes, well defined on each facet. Also, this allows us to naturally combine both the normal integration and normal estimation when these were two separate stages in most of previous approaches [13–15].

1.3 Contributions

This article presents several improvements to state-of-the-art 3D reconstruction techniques exploiting Helmholtz reciprocity. First, we model the problem as an energy minimization problem which is completely surface-based (when all previous methods were camera based). Secondly, we present a method for solving this problem and show how to implement it on triangular surface meshes by using a coherent discrete gradient flow. Finally, since we optimize a full surface, the method is able handle visibility. This also allows to fully reconstruct dense 3D surfaces of complex objects in a single framework.

2 Helmholtz Stereopsis - Variational Formulation

As described previously, Helmholtz reciprocity exploits the fact that BRDFs are generally symmetric and therefore, for any incoming angle $\hat{i}$ and outgoing direction $\hat{o}$, we have $\beta(\hat{i}, \hat{o}) = \beta(\hat{o}, \hat{i})$ where β is the BRDF function. By interchanging light and camera positions, one can exploit the constraint on β in radiance equations in order to solve 3D reconstruction problems.

Given a camera – light pair, one can write the radiance equation I_c of a scene seen from a camera c as:

$$I_c = \alpha\beta(\mathbf{v}_c, \mathbf{v}_l)\frac{\mathbf{v}_l \cdot \mathbf{n}}{|\mathbf{v}_l|^3} \,, \tag{1}$$

where $\mathbf{v}_c$ is the vector from the camera center to the point, $\mathbf{v}_l$ is the vector from the light center to the view-point and α is a constant. $\mathbf{n}$ is the surface normal of the considered point and $\beta(\mathbf{v}_c, \mathbf{v}_l)$ is the BRDF at the surface point (See Figure 2). The same radiance equation can be written for modeling the radiance I_l with the BRDF $\beta(\mathbf{v}_l, \mathbf{v}_c)$. Using Helmholtz reciprocity allows us to write $\beta(\mathbf{v}_c, \mathbf{v}_l) = \beta(\mathbf{v}_l, \mathbf{v}_c)$. This equality then defines the Helmholtz stereopsis constraint for all the point of the surface $\mathcal{S}$:

$$\left[I_c \, \frac{\mathbf{v}_c}{|\mathbf{v}_c|^3} - I_l \, \frac{\mathbf{v}_l}{|\mathbf{v}_l|^3} \right] \cdot \mathbf{n} = 0 \,. \tag{2}$$

Now, we are going to formulate this constraint in the variational framework via a weighted area functional defined over the surface of the object.
We denote by $\pi_c(\mathbf{x})$ (resp. $\pi_l(\mathbf{x})$) the projection of a point in space $\mathbf{x}$ in the camera c (light l respectively), and I_c (or I_l) its corresponding intensity value in the image. For more clarity, we also denote

$$\mathbf{h}(\mathbf{x}) = \left[I_c(\pi(\mathbf{x}))\frac{\mathbf{v}_c}{|\mathbf{v}_c|^3} - I_l(\pi(\mathbf{x}))\frac{\mathbf{v}_l}{|\mathbf{v}_l|^3} \right] \,. \tag{3}$$

In this case, the surface that "best" verifies Equation (2) can be obtained by minimizing the following energy functional defined over the surface, with respect to the surface itself:

$$E_{HS}(\mathcal{S}) = \int_{\mathcal{S}} \left(\mathbf{h}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})\right)^2 \; \nu_{\mathcal{S},c,l}(\mathbf{x}) \, \mathrm{d}s \,, \tag{4}$$

where $\nu_{\mathcal{S},c,l}$ is the characteristic function such that $\nu_{\mathcal{S},c,l}(\mathbf{x}) = 1$ if $\mathbf{x}$ is visible from both images, or 0 otherwise. $\mathrm{d}s$ is the element of area of the surface. This problem formulation allows thus to naturally integrate both multiview geometry and normal constraint. The functional (4) constrains the surface normals to be on the orthogonal plane of $\mathbf{h}(\mathbf{x})$. This is an ill-posed problem since there is an unlimited choice for the normal. In this context several reciprocal pairs are needed in order to better pose the problem and Energy (4) has to be adapted to multiview settings. At the end, we then consider the energy to minimize as the sum of all energies for all Helmholtz pairs i:

$$E_{HS}(\mathcal{S}) = \sum_i \int_{\mathcal{S}} \left(\mathbf{h}_i(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})\right)^2 \; \nu_{\mathcal{S},c_i,l_i}(\mathbf{x}) \, \mathrm{d}s \,, \tag{5}$$

where i is the ith camera / light pair.

In the next section, we show how to minimize this energy via gradient descent when the surface is represented by a triangle mesh. To simplify the notations, we will just consider and compute the gradient of the functional (4). The gradient of (5) is then obtained by summing the gradients of all the camera / light pairs.

3 Optimization for Triangle Mesh Representation

3.1 Choice of Representation

In section 1.2 we show the interest of using an intrinsic and full surface representation. In practice, there exists several possibilities for such representation. We propose to minimize energy (5) using a gradient descent algorithm. The choice of the representation must then be consistent with the surface evolution technique which is the base of gradient descent methods. Here we chose to use the Lagrangian framework and to represent the surface by a triangle mesh. Several reasons motivate this choice. In recent years, Lagrangian methods have taken advantage of significant advances in mesh processing allowing these methods to enjoy practical properties such as topological changes [18, 19]. In Lagrangian methods, the gradient is computed directly from the discrete representation, whereas in the Eulerian framework the continuous gradient is computed and then discretized. Performing gradient descent in the context of discrete representations allows to make the minimization coherent with the handled numerical object. In other respects, the visibility of a point from a vantage point is well defined and easy to compute with a mesh representation (by using graphic hardware). In practice, it is easier to check the visibility with such a representation than with a level-set representation. All these reasons make nowadays Lagrangian methods more and more popular. Also these methods have recently proved their strong potential for 3D applications [8, 20, 21].

3.2 Shape Gradient and Evolution Algorithm

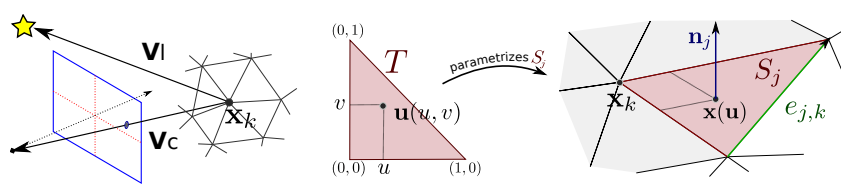


Fig. 2. Notations. Parametrization of the discrete representation of the surface into a triangle mesh.

From now, we consider that the surface is a piecewise planar triangular mesh. Let $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ be a discrete mesh, $\mathbf{x}_k$ being the kth vertex of $\mathbf{X}$, and let $\mathcal{S}_j$ be the jth triangle of $\mathbf{X}$. With such a representation, functional (4) can be rewritten as:

$$E_{HS}(\mathcal{S}) = \sum_j \int_{\mathcal{S}_j} \left(\mathbf{h}(\mathbf{x}) \cdot \mathbf{n}_j\right)^2 \, \nu_{\mathcal{S},c,l}(\mathbf{x}) \, \mathrm{d}s_j \ , \tag{6}$$

where $\mathbf{n}_j$ is the normal to $\mathcal{S}_j$ and where the sum is over all the triangles of the mesh $\mathbf{X}$. Figure 2 illustrates these notations.

We propose to optimize E_{HS} with respect to $\mathcal{S}$ using the shape gradient [22]. Let $\mathbf{V}$ be a vector field defined on all the vertices $\mathbf{x}$ of the mesh $\mathbf{X}$ representing the surface deformation. Let us consider the evolution of E_{HS} according to the deformation $\mathbf{V}$. In other words, we assume that the vertices $\mathbf{x}_k[t]$ of $\mathbf{X}[t]$ are moving according to $\mathbf{x}_k[t] = \mathbf{x}_k^0 + t\mathbf{V}_k$. The method for computing the gradient of E_{HS} with respect to $\mathcal{S}$ consists in computing the directional derivative of $E(\mathcal{S})$ for this deformation $\mathbf{V}$, i.e., $\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0}$, and then in rewriting it as a scalar product of $\mathbf{V}$, i.e. as $\langle \mathbf{V}, \mathbf{G}\rangle = \sum_k \mathbf{V}_k \cdot \mathbf{G}_k$. The obtained vector $\mathbf{G}$ is called the *gradient* and the energy necessarily decreases when deforming the surface according to its opposite direction $-G$. Indeed, for $\mathbf{x}_k[t] = \mathbf{x}_k^0 - t\mathbf{G}$, we have $\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = -\langle \mathbf{G}, \mathbf{G}\rangle \leq 0$, see [20].

To obtain the gradient of our energy E_{HS}, we have then to calculate the expression of $\frac{d}{dt}E_{HS}(\mathcal{S}[t])\Big|_{t=0}$ and express it as a scalar product of $\mathbf{V}$. In appendix A, we have detailed this calculus in the general case where a functional $\int_{\mathcal{S}} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\mathrm{d}s$ is minimized. By replacing $g(\mathbf{x}, \mathbf{n}(\mathbf{x}))$ by $(\mathbf{h}_i(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}))^2$, we get:

$$\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = \sum_j \sum_{k\in \mathcal{K}_j} \mathbf{V}_k \cdot \left\{ \frac{\mathbf{e}_{j,k}}{A_j} \wedge \int_{\mathcal{S}_j} \Big((\mathbf{h}\cdot\mathbf{n}_j)^2 \mathbf{n}_j - 2\,(\mathbf{h}\cdot\mathbf{n}_j)\,\mathbf{h} \Big)\,\mathrm{d}s_j \right.$$
$$\left. - \int_{\mathcal{S}_j} 2(\mathbf{h}\cdot\mathbf{n}_j)\nabla_{\mathbf{x}}(\mathbf{h}\cdot\mathbf{n}_j)\,\phi_k(\mathbf{x})\,\mathrm{d}s_j \right\}, \quad (7)$$

where A_j is the area of $\mathcal{S}_j$, $\mathbf{e}_{j,k}$ is the edge of $\mathcal{S}_j$ that is at the opposite of vertex k; $\mathcal{K}_j$ is the set of the indexes of the three vertices of the triangle $\mathcal{S}_j$ and $\phi_k : \mathcal{S} \to \mathbb{R}$ is the piecewise linear interpolating basis function such that $\phi_k(\mathbf{x}_k) = 1$ and $\phi_k(\mathbf{x}_i) = 0$ if $i \neq k$.
Then the L^2 gradient descent flow using the triangular mesh uses the gradient $\nabla E(\mathbf{X}) = M^{-1}\dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X})$, where M is the *mass* matrix and $\dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X})$ is directly given by the part in braces of Equation (7). One classically approximates M by the diagonal mass lumping $\tilde{M}$, where $\tilde{M}_{ii}$ is the area of the Voronoi dual cell of $\mathbf{x}_i$ times the identity matrix Id_3; for more details, see e.g. [20, 23]. The evolution algorithm used here is:

$$\begin{cases} \mathbf{X}[0] = \mathbf{X}^0\,, \\ \mathbf{X}[t+1] = \mathbf{X}[t] - dt\,\tilde{M}^{-1}\dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X}[t])\,, \end{cases} \quad (8)$$

where $\mathbf{X}^0$ is some initial mesh.

4 Experimental Results

In this section we present results using the gradient developed in section 3 (Equation (7)) that directly corresponds to the minimization of our original Energy (Equation (6)).

Let us emphasize here that the gradient descent flow used to obtain the result is exactly the one described previously. In particular, it does not require additional terms or parameters such as surface smoothing present in most variational formulations. Of course, adding such a term would help in being more robust to noise and calibration errors. The only parameters used by us are for numerical computations, like for instance the numerical integration over the triangles. All those can be easily estimated automatically. Since this is a gradient descent approach, one needs some reasonable initialization of the surface such as the visual hull to avoid local minima.

The experiments were implemented in C++ and OpenGL using the CGAL library for mesh computation running on a standard 2.4GHz linux machine – and the topology adaptive meshes of [18]. The optimization starts from an initial condition which is the visual hull in our case. A coarse-to-fine approach is applied to help prevent from local minima. The rendered results use only one constant normal per facet (flat shading).

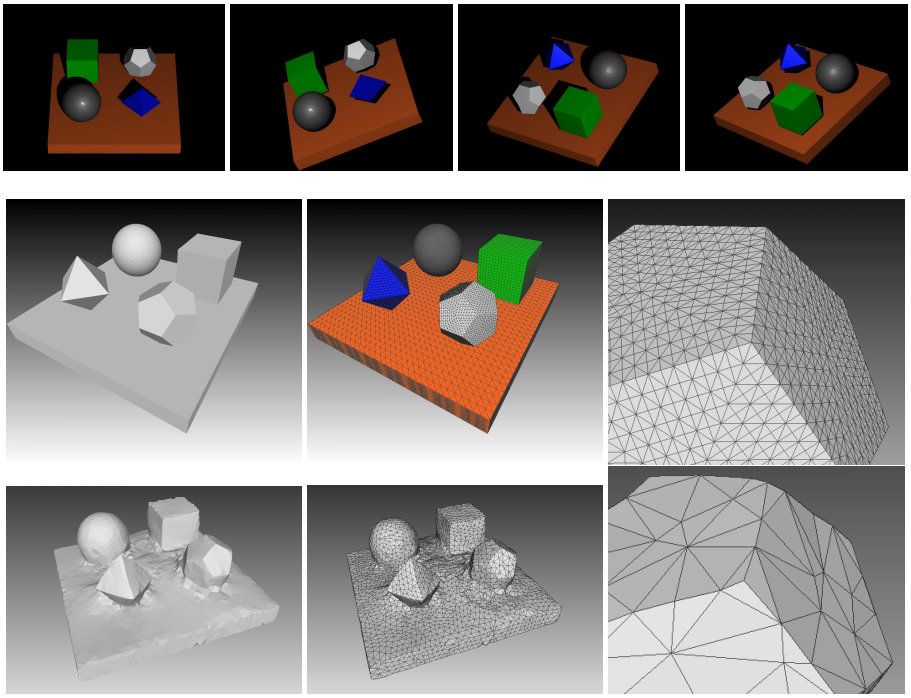


Fig. 3. A simple synthetic example. Top row: 2 camera / light pairs out of the 32. Middle row: ground truth surface; ground truth mesh representation; Bottom row: final result; mesh representation of the result; details on the mesh.

We first apply our method to synthesized data. Figure 3 show an example of simple objects disposed on a plane, where images were generated using non-Lambertian reflectance. This dataset is composed of 32 reciprocal pairs with images of resolution 800×600, placed all around the object of interest. This example shows that our method

is able to recover the surface whereas previous Helmholtz stereo methods, where visibility is not accounted for, would fail. In order to solve this problem, they would need to cluster the camera positions to find several view-centered cameras, integrate multiple normal maps and then merge the final reconstructions into a single surface. Our method is simpler in the sense that it works without additional steps.

Even though the number of vertices is low, the gradient flows tends to place them in their correct location. In particular triangle edges perfectly match the one in the images. Again, this is made possible because the discrete gradient is computed with respect to the discrete representation. Also the approach is suitable for reconstructing objects with sharp edges, having depth discontinuities or self-occlusions.

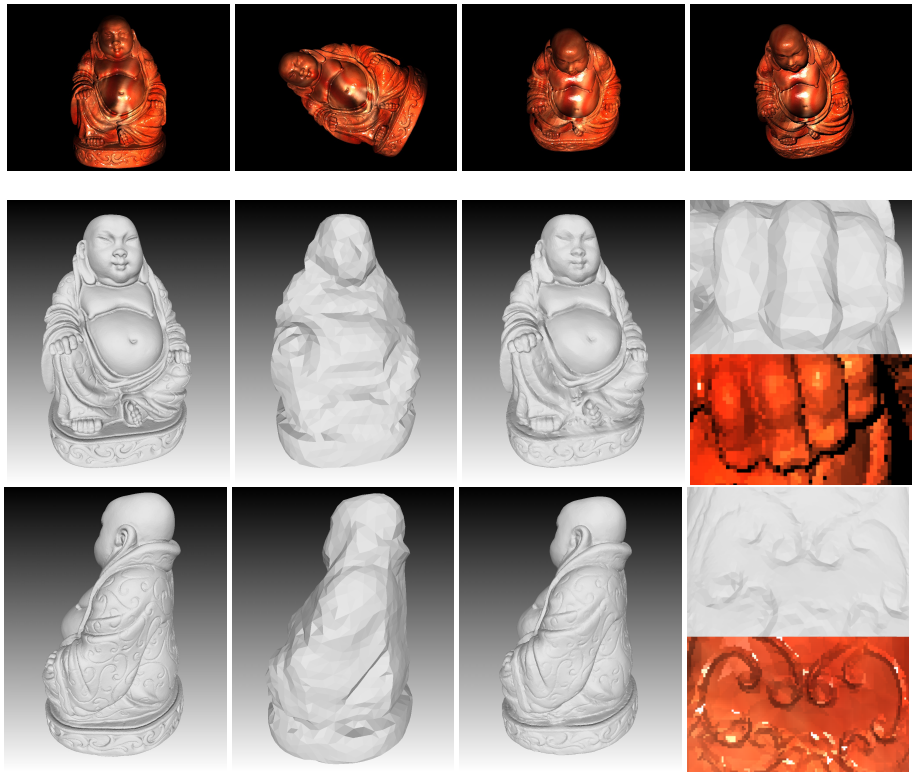


Fig. 4. Buddha dataset. Top row: 2 camera – light pairs out of the 32. Bottom row: ground truth surface; initial visual hull; estimated mesh; input image zoom and corresponding recovered mesh details.

The example in figure 4 shows that our method can be applied to reconstruct full and high quality object shapes. Even though the input resolution of the images is low (1024×768), the recovered surface nicely matches the ground truth model. The images were generated using a mixture of different specular models so that it looks realistic and

non Lambertian. Details are well recovered, and the quality of the mesh is good enough for further object relighting.

Following the evaluation presented in [24], we perform a quantitative evaluation of examples from Figures 3 and 4 where we consider the object in a 2m diameter bounding box. For Figure 3, the completeness at 10mm is 87.51%, and the accuracy at 95% is 9.301mm. For the Buddha, the completeness at 10mm is 95.766%, and the accuracy at 95% is 5.29mm.

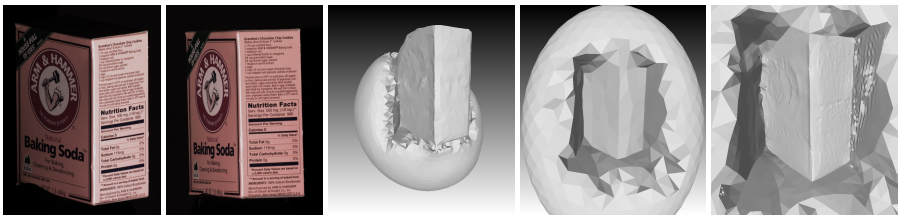


Fig. 5. Small box data: results of the proposed method with different initial conditions and mesh resolution. From left to right: 1 camera – light pair out of the 8 pairs; result from a small non-encompassing initial surface; result from an encompassing initial surface; result from an encompassing initial surface with a more dense resolution.

The next two figures 5 and 6 show real data used in previous work [13, 14], that have been capture only from one side. We then cannot reconstruct the full surface of the object since images behind the object are missing. Figure 5 shows the 3D reconstruction of a textured box. Since we can choose the mesh resolution, having large triangles compared to the image resolution allows to integrate the gradient over the triangle and have a correct gradient flow for the vertices of the triangle. Then we can reconstruct objects with textured or rough surfaces using Helmholtz reciprocity similarly as in [15]. This figure also show different optimization taken from different initial conditions that finally give similar results.

Figure 6 show results for two real datasets, the mask containing 18 reciprocal pairs, and the mannequin containing 8 reciprocal pairs.

Finally, the approach was tested on real dataset on full 3D objects (Figure 7). It consists of 18 reciprocal pairs (using 1104×828 resolution images) taken on a ring around the object slightly on top of it. The two data show a "Fish" reconstruction highly specular with fine changing surface structure, whereas the second one "Dragon" has strong self-occlusions and complex shape. Starting from the visual hull, we are able to recover details on the surface even though the input image resolution is not too high. Results are illustrated in Figure 7 and present the recovered full 3D surface. Some parts of the surface are not visible from the images and thus cannot be recovered. Camera calibration was performed using a checker board without distortion corrections and light positions were empirically positioned and calibrated. These datasets will be made available for comparisons purposes. Images are taken around the object so some parts are occluded from images and objects also contain self-occlusions. For those reasons,

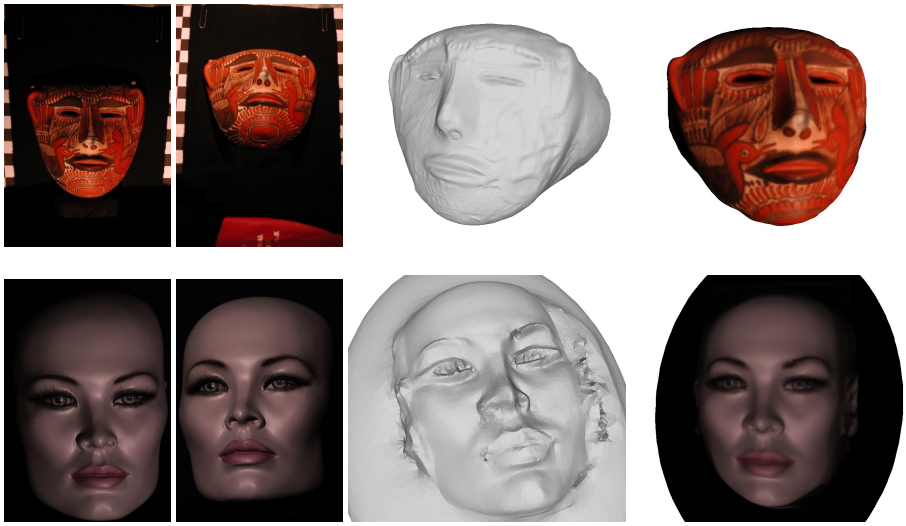


Fig. 6. Mannequin (8 reciprocal pairs) and Mask (18 reciprocal pairs) data: results of the proposed method for two real dataset containing varying complex appearance. From left to right: 1 camera / light pair out of the 18 (respectively 8); final result; final mesh textured with the mean of the reprojected colors from the cameras.

previous approaches using a camera-center view do not apply – since it requires depth continuity – whereas our surface-based approach can recover the full 3D shape.

4.1 Discussion

All these examples illustrated the advantage of having a mesh representation. It allows to preserve edges, and we show that having a coherent discrete gradient flow allows vertices to be placed at their correct locations. Possible extensions to the approach may use automatic adaptive meshes, where big triangles would fill planar surface parts and high curvature would have a more dense mesh.

Second order minimization is known to recover well the higher frequency of the surface, but poorly recover the lower ones. Also, due to the integration process, the optimization might be slow if the surface is too far from the solution. A good initial condition and a coarse to fine approach significantly help to prevent these problems, as illustrated in the experiments. In particular, the gradient flow tends to shrink the surface and introduce a minimal surface bias.

However, since the method minimizes a weighted area functional defined over the surface, one of its global minima (in addition to the real surface) is the empty set. To prevent from this choice, one can add an additional term, start closer to the solution, or fix boundary conditions. A more elegant way is to see the problem as a reprojection error. Instead of minimizing a weighted area functional, one can reformulate the problem

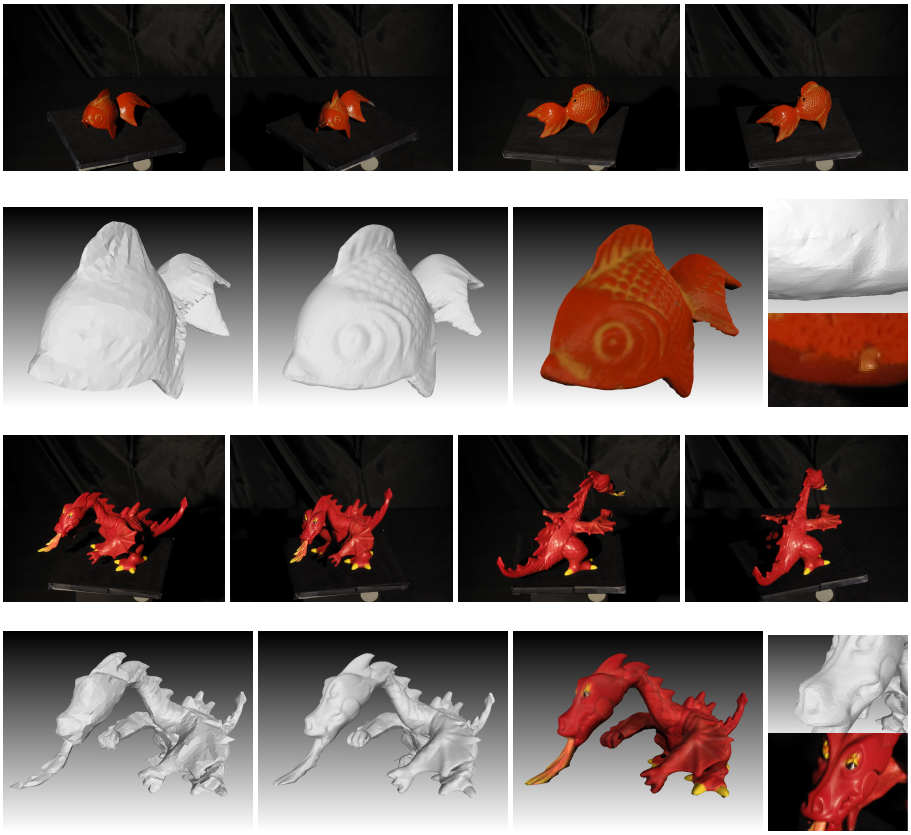


Fig. 7. Fish and Dragon data (18 reciprocal pairs of 1104×828): Results of the proposed method for two full 3D real world datasets. Top and third row: 2 camera / light pairs out of the 36. Second and last row: initial visual hull; recovered 3D mesh; final mesh textured with the mean of the reprojected colors from the cameras; input image zoom and corresponding recovered mesh details.

(2) by minimizing the following energy functional:

$$E(\mathcal{S}) = \int_{\mathcal{I}_c} \left(I_c - \left(I_l \circ \pi_{\mathcal{S},I_l} \circ \pi^{-1}_{\mathcal{S},I_c} \right) \frac{\hat{\mathbf{v}}_l \cdot \left(\mathbf{n} \circ \pi^{-1}_{\mathcal{S},I_c} \right)}{\hat{\mathbf{v}}_c \cdot \left(\mathbf{n} \circ \pi^{-1}_{\mathcal{S},I_c} \right)} \right)^2 \mathrm{d}\mathbf{u} \,, \tag{9}$$

where $\pi^{-1}_{\mathcal{S},I_c}(\mathbf{u})$ is the reprojection of an image point $\mathbf{u}$ of the image I_c on the surface $\mathcal{S}$. This formulation as been presented by [17], but has not been minimized as a reprojection error, like for instance in the case of [25] for continuous surfaces, and of [21] using deformable meshes. In fact one may rewrite Equation (9) as an energy over the visible surface [25] instead of the image in order to optimize the surface. Such formulation lets appear an additional term, that turns out to behave like a visual hull constrain on the

silhouette occluding contours, or as a contour matching term for self occlusions. Such a term will give boundary conditions to prevent from shrinkage, and will help the method to be more robust to initial conditions.

5 Conclusion

In this paper we have presented a surface-based method to estimate the 3D shape of objects from multiview Helmholtz stereo pairs. As far as we know, this is the first time Helmholtz stereopsis can be used to recover dense and full 3D models into a single framework. This is made possible thanks to the compact surface representation that allow to easily compute surface point visibility. Moreover, the mesh based representation allows to naturally exploit geometric relationships between a point of the scene and its surface normal. Tests on synthetic and real datasets demonstrate the benefit of our approach.

Acknowledgement. Research was supported by the Agence Nationale pour la Recherche within the Flamenco project (Grant ANR-06-MDCA-007). The authors would like to thank Todd Zickler for providing Helmholtz stereo data used in Figures 5 and 6.

References

1. Goesele, M., Curless, B., Seitz, S.M.: Multi-view stereo revisited. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 02. (2006) 2402–2409
2. Jin, H., Cremers, D., Yezzi, A.J., Soatto, S.: Shedding light on stereoscopic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 01. (2004) 36–42
3. Jin, H., Cremers, D., Wang, D., Prados, E., Yezzi, A., Soatto, S.: 3-d reconstruction of shaded objects from multiple images under unknown illumination. International Journal of Computer Vision **76** (2008)
4. Hernandez, C., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008) 548–554
5. Mallick, S.P., Zickler, T., Kriegman, D.J., Belhumeur, P.N.: Beyond lambert: Reconstructing specular surfaces using color. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 619–626
6. Zickler, T., Mallick, S.P., Kriegman, D.J., Belhumeur, P.: Color subspaces as photometric invariants. International Journal of Computer Vision (2008)
7. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. The International Journal of Computer Vision **72** (2007) 179–193
8. Vu, H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo. In: IEEE Conference on Computer Vision and Pattern Recognition. (2009)
9. Jin, H., Soatto, S., Yezzi, A.J.: Multi-view stereo beyond lambert. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2003) 171
10. Yu, T., Xu, N., Ahuja, N.: Recovering shape and reflectance model of non-lambertian objects from multiple views. In: IEEE Conference on Computer Vision and Pattern Recognition. (2004) 226–233
11. Yoon, K.J., Prados, E., Sturm, P.: Joint estimation of shape and reflectance using multiple images with known illumination conditions. International Journal of Computer Vision (2009) to appear.

12. Hertzmann, A., Seitz, S.: Example-based photometric stereo: Shape reconstruction with general, varying brdfs. IEEE Trans. Pattern Anal. Mach. Intell. **27** (2005) 1254–1264
13. Zickler, T., Belhumeur, P.N., Kriegman, D.J.: Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. International Journal of Computer Vision **49** (2002) 215–227
14. Zickler, T., Ho, J., Kriegman, D., Ponce, J., Belhumeur, P.: Binocular helmholtz stereopsis. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. (2003) 1411–1417 vol.2
15. Guillemaut, J.Y., Drbohlav, O., Sara, R., Illingworth, J.: Helmholtz stereopsis on rough and strongly textured surfaces. In: 3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium, Washington, DC, USA, IEEE Computer Society (2004) 10–17
16. Zickler, T.: Reciprocal image features for uncalibrated helmholtz stereopsis. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 2. (2006) 1801–1808
17. Tu, P., Mendon, P.R.S.: Surface reconstruction via helmholtz reciprocity with a single image pair. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **1** (2003) 541
18. Pons, J.P., Boissonnat, J.D.: Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA (2007)
19. Zaharescu, A., Boyer, E., Horaud, R.: Transformesh: a topology-adaptive mesh-based approach to surface evolution. In: Proceedings Asian Conference on Computer Vision, Tokyo, Japan (2007)
20. Eckstein, I., Pons, J.P., Tong, Y., Kuo, C.C.J., Desbrun, M.: Generalized surface flows for mesh processing. In: Eurographics Symposium on Geometry Processing. (2007)
21. Delaunoy, A., Prados, E., Gargallo, P., Pons, J.P., Sturm, P.: Minimizing the multi-view stereo reprojection error for triangular surface meshes. In: British Machine and Vision Conference, Leeds, UK. (2008)
22. Debreuve, ., Gastaud, M., Barlaud, M., Aubert, G.: Using the shape gradient for active contour segmentation: from the continuous to the discrete formulation. Journal of Mathematical Imaging and Vision (2007)
23. Meyer, M., Desbrun, M., Schröder, P., Barr, A.H.: Discrete differential-geometry operators for triangulated 2-manifolds. (2002)
24. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition. (2006) 519–528
25. Gargallo, P.: Contributions to the Bayesian approach to Multi-view Stereo. PhD thesis, Institut National Polytechique de Grenoble, France (2008)

A Gradient Flows of Weighted Area Functionals using Triangular Meshes

Let $\mathcal{S}_j$ be the jth triangle of the mesh and let us consider a parametrization of the triangle $\mathcal{S}_j$ such that $\mathbf{x}(\mathbf{u}) = x_k^j + u\,\overrightarrow{x_k^j x_1^j} + v\,\overrightarrow{x_k^j x_2^j}$ where x_k^j and x_1^j and x_2^j are the three vertices associated to the triangle $\mathcal{S}_j$ and where $\mathbf{u}(u,v) \in T = \{(u,v) | u \in [0,1] \text{ and } v \in [0,u]\}$. (See Figure 2)
In this appendix, we detail the calculus of $\frac{d}{dt} E(\mathcal{S}_j[t])\Big|_{t=0}$ for energy functionals of the form:

$$E(\mathcal{S}) = \sum_i \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}(\mathbf{x}))\, d\sigma = \sum_j 2\, A_j \int_T g(\mathbf{x}(\mathbf{u}), \mathbf{n}_j)\, \mathrm{d}\mathbf{u}\,, \tag{10}$$

where A_j is then the area of the triangle $\mathcal{S}_j$ and $\mathbf{n}_j$ the outward surface normal of triangle $\mathcal{S}_j$. Let us focus on the evolution of $E(S)$ under the induced velocity $\mathbf{V}$ on triangle $\mathcal{S}_j$ only (In other words when a vertex $\mathbf{x}_k$ is moved according to $\mathbf{x}_k[t] = \mathbf{x}_k^0 + t\mathbf{V}_k$). We have:

$$E(\mathcal{S}_j[t]) = 2\,A_j[t] \int_T g(\mathbf{x}(\mathbf{u}) + t\mathbf{V}(\mathbf{x}(\mathbf{u})), \mathbf{n}_j[t])\,\mathrm{d}\mathbf{u}\,, \tag{11}$$

where $A_j[t]$ is the area of the triangle $\mathcal{S}_j[t]$, $\mathbf{n}_j[t]$ is its normal and $\mathbf{V}(\mathbf{x})$ is the piecewise linear extension of the $\mathbf{V}$ vector field on the whole surface $\mathcal{S}$. By simple derivation we get:

$$\begin{aligned}\frac{d}{dt}E(\mathcal{S}_j[t])\Big|_{t=0} = \frac{A_j'[0]}{A_j} \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s_j \\ + \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j)\cdot \mathbf{V}(\mathbf{x})\,\mathrm{d}s_j \;+ \int_{\mathcal{S}_j} \nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j)\cdot \mathbf{n}_j'[0]\,\mathrm{d}s_j\,,\end{aligned} \tag{12}$$

where $A_j'[0] = \frac{d}{dt}A_j[t]\Big|_{t=0}$ and $\mathbf{n}_j'[0] = \frac{d}{dt}\mathbf{n}_j[t]\Big|_{t=0}$ are detailed below. In order to rewrite $\frac{d}{dt}E(\mathcal{S}_j[t])\Big|_{t=0}$ as a scalar production of $\mathbf{V}$, we then need to detail $A_j'[0]$ and $\mathbf{n}_j'[0]$. Since for all $\mathbf{x} \in \mathcal{S}_j$, $\mathbf{V}(\mathbf{x}) = \sum_{k\in K_j} \mathbf{V}_k \phi_k(\mathbf{x})$ (K_j, ϕ_k and $\mathbf{e}_{j,k}$ being defined in section 3.2), one can show that: $A_j'[0] = \sum_{k\in K_j} \frac{1}{2}\,(\mathbf{n}_j \wedge \mathbf{e}_{j,k})\cdot \mathbf{V}_k$, and

$$\mathbf{n}_j'[0] = \frac{1}{2\,A_j}\left(\left(\sum_{k\in K_j} \mathbf{e}_{j,k}\wedge \mathbf{V}_k\right) - \left((\sum_{k\in K_j} \mathbf{e}_{j,k}\wedge \mathbf{V}_k)\cdot \mathbf{n}_j\right)\mathbf{n}_j\right)\,. \tag{13}$$

It follows that

$$\begin{aligned}\frac{d}{dt}E(\mathcal{S}_j[t])\Big|_{t=0} = \sum_{k\in K_j} \mathbf{V}_k\cdot \Bigg\{ \frac{\mathbf{n}_j\wedge \mathbf{e}_{j,k}}{2A_j} \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s_j \\ + \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j)\phi_k(\mathbf{x})\,\mathrm{d}s_j - \frac{\mathbf{e}_{j,k}}{2\,A_j} \wedge \int_{\mathcal{S}_j} g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s_j \Bigg\}.\end{aligned} \tag{14}$$

Above, we have denoted $g_{\mathbf{n}} = \nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j) - \langle \nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j), \mathbf{n}_j\rangle\ \mathbf{n}_j$, where $\nabla_{\mathbf{n}} g(\mathbf{x}, \mathbf{n}_j)$ is the gradient of g with respect to the second variable (i.e. $\mathbf{n} \in \mathbb{R}^3$).

It then immediately follows that:

$$\begin{aligned}\frac{d}{dt}E(\mathcal{S}[t])\Big|_{t=0} = \sum_k \mathbf{V}_k \cdot \sum_{j\in \mathcal{J}_k} \Bigg\{ \int_{\mathcal{S}_j} \nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j)\phi_k(\mathbf{x})\,\mathrm{d}s_j \\ - \frac{\mathbf{e}_{j,k}}{2A_j} \wedge \int_{\mathcal{S}_j} g(\mathbf{x}, \mathbf{n}_j)\mathbf{n}_j + g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j)\,\mathrm{d}s_j \Bigg\},\end{aligned} \tag{15}$$

where $\mathcal{J}_k$ is the set of triangles containing vertex $\mathbf{x}_k$. By replacing $g(\mathbf{x}, \mathbf{n}(\mathbf{x}))$ by $(\mathbf{h}_i(\mathbf{x})\cdot \mathbf{n}(\mathbf{x}))^2$ in the equations below, where $\mathbf{h}_i(\mathbf{x})$ is the vector defined in the paper, we have:

$$\begin{aligned}\nabla_{\mathbf{x}} g(\mathbf{x}, \mathbf{n}_j) &= 2(\mathbf{h}(\mathbf{x})\cdot \mathbf{n}_j)\nabla_{\mathbf{x}}(\mathbf{h}(\mathbf{x})\cdot \mathbf{n}_j)\,, \\ g(\mathbf{x}, \mathbf{n}_j)\mathbf{n}_j &= (\mathbf{h}(\mathbf{x})\cdot \mathbf{n}_j)^2\,\mathbf{n}_j\,, \\ \text{and}\ \ g_{\mathbf{n}}(\mathbf{x}, \mathbf{n}_j) &= 2\,(\mathbf{h}(\mathbf{x})\cdot \mathbf{n}_j)\,\mathbf{h}(\mathbf{x}) - 2\,(\mathbf{h}(\mathbf{x})\cdot \mathbf{n}_j)^2\,\mathbf{n}_j\,.\end{aligned} \tag{16}$$

A Non-Local Approach to Shape From Ambient Shading

Emmanuel Prados[1], Nitin Jindal[1], and Stefano Soatto[2]

[1] Perception Lab., INRIA Grenoble - Rhône-Alpes, FRANCE
[2] Computer Science Department, University of California, Los Angeles, USA

Abstract. We study the mathematical and numerical aspects of the estimation of the 3-D shape of a Lambertian scene seen under diffuse illumination. This problem is known as "shape from ambient shading" (SFAS), and its solution consists of integrating a strongly non-local and non-linear Integro-Partial Differential Equation (I-PDE). We provide a first analysis of this global I-PDE, whereas previous work had focused on a local version that ignored effects such as occlusion of the light field. We also design an original approximation scheme which, following Barles and Souganidis' theory, ensures the correctness of the numerical approximations, and discuss about some numerical issues.

1 Introduction

Shape From Shading (SFS) refers to the problem of computing the three-dimensional shape of a surface, under certain assumptions on its reflectance and on the illumination, from a single grayscale image. By necessity, to render the problem tractable, these assumptions are rather coarse: Most restrict the illumination to a single point-light source at infinity [22, 4, 15, 8]. Only recently, [16] have shown that the problem actually simplifies when the attenuation of the light source at finite distance is taken into account. Nevertheless, due to inter-reflections and other complex phenomena, modeling illumination as a point source is very unrealistic even on a bright sunny day. Indeed, in most realistic conditions including indoors and outdoor overcast conditions, a uniform hemispherical illumination source is a more realistic model. The study of SFS under such illumination conditions has been pioneered by Langer et al. [11, 18, 10], and followed by others that we discuss shortly. In this work, we focus on the mathematical properties of the problem of "Shape From Ambient Shading" (SFAS), and seek for conditions that render the problem well-posed.

The paper is organized as follows. In section 2, we describe the modeling assumptions and introduce our notation. In section 3 we formulate the Shape From Ambient Shading problem as an integro-differential equation. We develop our theoretical results in section 4. We describe our numerical algorithm in section 5 and show some experimental results in section 6. Some discussions and future work are given in section 7.

1.1 Relation to Prior Work

Langer et al. [11, 18, 10] were the first to consider the case of *ambient lighting*, and to note that vignetting effects, far from being a nuisance, enable the inference of object shape similar to more traditional SFS, except for the added complication of the distributed source. In [19], Tian, Tsui and Yeung have proposed a *numerical* SFS algorithm for dealing with some non-punctual and multiple light sources (any combination of spherical, rectangular and cylindrical light sources). Following a more elaborate and physically motivated model of illumination, [14, 20, 10, 21] introduced methods to deal with interreflection. However, in none of these works [19, 11, 18, 10, 14, 20, 10] are the mathematical properties of the SFAS problem elucidated analytically. In particular, there are no results on the existence and uniqueness of solution for the ensuing global PDE.

At the opposite end of the spectrum, Lions, Rouy and Tourin [12] performed a *theoretical analysis* the SFS problem for multiple and continuous distributed light sources. As Tian, Tsui and Yeung [19], Lions, Rouy and Tourin neglect shadows (i.e. occlusions of the light sources by the surface itself); more specifically, they assume that for any fixed point x on the surface, all the light sources located on the hemisphere normal to the surface at x are visible from this point. This allows them to neglect the global nature of the equation, which in turn significantly simplifies the analysis.

As Langer et al. [11, 18, 10] we focus on *ambient lighting*. In their work, Langer et al. do not neglect the "shadows effect" and they model interreflections. They also underline the importance of ambient lighting in psychophysics. In this context, light comes from *all* directions and the assumption of Lions, Rouy and Tourin [12] is equivalent to assume that the solution is concave. Here, we do not want to limit ourself to concave objects. Therefore, Lions' constraints are far too restrictive.[3]. The necessity to consider these phenomena takes us to mathematicaly uncharted territories. To the best of our knowledge, we are the first to provide theoretical results for the SFAS problem. Also, we introduce numerical algorithms verifying the properties of monotony, consistency and stability which typically ensure its convergence (see [1]).

2 Modeling Shape From Ambient Shading

Shape From Shading exploits assumptions on the illumination and reflectance properties of the scene (or of an object of interest within the scene) to relate its three-dimensional (3-D) shape to the measured grayscale image. The most typical assumptions are that the scene is Lambertian with constant diffuse albedo. This is akin to chalk and rough stone, and neglects specularities, translucency and other complex phenomena in the interaction of light with matter. While this assumption is clearly violated in most natural and man-made scenes, there are significant portions of scenes where the assumption is reasonable, and even objects that are far from Lambertian, such as human faces, have been successfully approximated as such for the purpose of analysis and inference (but not for synthesis, as humans are evolutionarily atuned to discriminate subtle features in human faces). Clearly, being SFS an ill-posed problem, there is no way to validate the assumptions on the data themselves, so applying SFS to a scene that is not Lambertian and that does not have constant diffuse albedo will results in gross errors even if the SFS algorithm used is provably correct and optimal. The second class of assumptions commonly made concern illumination. The most common assumption, that of a point light source, is made more for mathematical convenience than for realism. Under this model, anything hidden from direct line-of-sight to the sun would be invisible, clearly a far cry from reality. Modeling the entire sky as a constant-radiance hemisphere seems to be equally crude, but indeed it has been shown to be a better approximation that a single point-light source [9]. Clearly, both phenomena are important and we auspicate their eventual integration. In the next subsection we formalize these assumptions and introduce our notation.

2.1 Reflectance Assumptions

Let $\mathcal{S}$ be a 2-D surface embedded in $\mathbb{R}^3$ that supports a bi-directional reflectance distribution function (BRDF) β with Lambertian reflection and constant diffuse albedo ρ. In other words, following [7], the BRDF at a point $p \in \mathcal{S}$ does not depend on the viewing direction ν_{px}, but only on the light source direction ν and on the position of the point itself $p \in \mathcal{S}$: $\beta(p; \nu_{px}, \nu) = \rho$. Because the intensity of the light source is not known, without loss of generality we can assume that the albedo to be equal to 1, and attribute the actual value to the light source.

[3] For simplicity, however, we also neglect interreflections, as Lions et al. [12] did, and we lump their contribution into the ambient illumination term, up to additive errors.

2.2 Lighting Assumptions

Definition 1. [Ambient illumination] *A power density distribution $R_L(\nu)d\nu$ defined for any ν in the unit sphere $\mathbb{S}^2$ of $\mathbb{R}^3$ is called* ambient illumination.

We assume the dominating sky principle [11], so we neglect inter-reflections and, for any point of the surface, consider only radiant energy coming from the (spherical) sky. To simplify the problem, later we will also assume that the ambient illumination is homogeneous, that is to say, that the power density distribution is constant. As explained in section 2.3, this assumption is required if we want to get rid of other contraints while still keeping the problem manageable. For convenience, we assume $R_L(\nu) = 1$.
Now, unlike most previous work, we want to model the effect of self-occlusions, whereby the light source is only partly visible at each point. We then need to introduce the

Definition 2. [Light visibility] *Let q be a point in $\mathbb{R}^3$. We call* visibility function *and we denote $\chi_{\mathcal{S}}(q;\nu)$ the indicator function of the directions $\nu \in \mathbb{S}^2$ from q that are not occluded by $\mathcal{S}$, i.e. the function $\chi_{\mathcal{S}}(q;\nu) = 1$ if $\{q + \lambda\nu, \lambda \in \mathbb{R}_+\} \cap \mathcal{S} = \phi$, otherwise, $\chi_{\mathcal{S}}(q;\nu) = 0$. The* visibility cone *at $q \in \mathbb{R}^3$ is*

$$\mathbf{C}_{\mathcal{S},q} = \{\nu \in \mathbb{S}^2 : \chi_{\mathcal{S}}(q;\nu) = 1.\}$$

The visibility function specifies if a point q is reached by the light ray of direction ν. If the 3-D point q sees the light ray, then $\chi_{\mathcal{S}}(q;\nu) = 1$, else if this light ray is occluded by another part of the scene $\mathcal{S}$ then $\chi_{\mathcal{S}}(q;\nu) = 0$. The visibility cone assembles all the visible rays from a point q.

2.3 Resulting Radiance

Given the assumptions above, the radiance of the surface at a point p is given by

$$R_{\mathcal{S}}(p) = \int_{\mathbb{S}^2} \chi_{\mathcal{S}}(p;\nu)\langle\nu,\nu_p\rangle R_L(\nu)\, d\nu = \int_{\mathbf{C}_{\mathcal{S},p}} \langle\nu,\nu_p\rangle\, R_L(\nu)\, d\nu, \tag{1}$$

where ν_p is the unit normal vector to the surface S at p, see [7]. Here, the surface is implicitly assumed to be smooth. This ensures that all the ligth rays visible from a point come from above its tangent plane (the tangent plane would not be defined otherwise). So, for all points p on $\mathcal{S}$, all the ligth rays visible from that point are included in the hemisphere defined by the normal ν_p to the surface at that point; that is to say $\mathbf{C}_{\mathcal{S},p} \subset Hemi_{\nu_p}$. Therefore $\forall\nu \in \mathbf{C}_{\mathcal{S},p}$, $\langle\nu,\nu_p\rangle \geq 0$. To underscore the fact that our model would not make sense with discontinuous image intensities, but also for mathematical convenience, we will enforce this property in the radiance as:

$$R_{\mathcal{S}}(p) = \int_{\mathbf{C}_{\mathcal{S},p}} \langle\nu,\nu_p\rangle^+\, R_L(\nu)\, d\nu. \tag{2}$$

where for all a in $\mathbb{R}$, $a^+ = a$ if $a \geq 0$ and $a^+ = 0$ else.
Already at this point one can immediatly see the difficulty introduced by self-occlusions, for the integration domain of (2) is restricted to the visibility cone $\mathbf{C}_{\mathcal{S},p}$, which directly depends of the global geometry of the scene $\mathcal{S}$. This is unlike traditional SFS, where the radiance only depended on local properties of the scene, for instance the direction of the normal ν_p to the surface at a given point. This requires the deployment of a different arsenal of tools that traditionally considered in SFS. [4]

[4] In order to simplify the problem and to remove this global dependency, Lions, Rouy and Tourin [12] assume that for all the points of the surface, *all the light sources* located on the normal hemisphere *are*

Unlike most prior work, we want to consider *full* ambient illumination $R_L(\nu)dL(\nu)$; i.e., we want to deal with R_L distribution such that $supp(R_L) = \mathbb{S}^2$. In such a case, the assumptions of [12] are equivalent to assuming that the surface is convex which is too restrictive an assumption. On the other hand, to simplify the notation and keep the problem tractable, we assume that the *ambient illumination is uniform*, i.e. $R_L(\nu)dL(\nu) = R_0 dL$, where $R_0 \in \mathbb{R}$. For convenience, we assume that $R_0 = 1$. In the next section we relate the measurements, i.e. the image greyscale, to the unknown – the 3-D shape of the scene – via the model above.

3 Shape from Ambient Shading

In this section we formalize the problem of SFAS as the solution of a global integro-partial differential equation, which we analyze in the next section.

3.1 Imaging Equation

We assume that we measure a greyscale image $I : \mathfrak{D} \subset \mathbb{R}^2 \to \mathbb{R}^+; x \mapsto I(x)$, on a closed domain $\mathfrak{D}$. Our goal is to characterize the surfaces $\mathcal{S}$ which generate it. Note that in general there is no guarantee that the surface is unique. We now need to link the measurements (I) with the unknowns ($\mathcal{S}$). To do so we use the assumptions developed in the previous section, together with the so-called Radiance equation [7], which approximates the brightness of a pixel x of the image with the radiance of the point $\pi_{\mathcal{S}}^{-1}(x)$ of the surface viewed in x: $I(x) = R_{\mathcal{S}}(\pi_{\mathcal{S}}^{-1}(x))$. Using the results from the previous section we have

$$I(x) = \int_{\mathbf{C}_{\mathcal{S},p}} \langle \nu, \nu_p \rangle^+ \, d\nu, \tag{3}$$

where ν_p is the outward-pointing normal vector to the surface $\mathcal{S}$ at the point $p = \pi_{\mathcal{S}}^{-1}(x)$. In what follows we are going to assume that the data I corresponds with an image of a scene verifying our modeling assumptions. In particular, for convenience, we rescale the range so as to have $0 \leq I(x) \leq \pi$. Also for simplicity, we assume that the camera performs an *orthographic projection* of the scene. This is a reasonable hypothesis provided that the domain of interest in the scene is small compared to its distance to the camera. Under these conditions, we can represent the surface as the graph of a function u, and write the outward unit normal vector explicitly:

$$\mathcal{S} = \{(x, u(x));\ x \in \mathfrak{D}\}\,; \qquad \nu_{(x,u(x))} = \frac{1}{\sqrt{1 + |\nabla u(x)|^2}}(-\nabla u(x), 1).$$

Finally, following [15], we could assume that the camera is a pinhole. This assumption could be forgone at the cost of a more complicated notation, but the core of the analysis in this paper would hold nevertheless.

visible. More specifically, they assume that $supp(R_L) \cap Hemi_{\nu_p} \subset \mathbf{C}_{\mathcal{S},p}$ where for a vector $\mathbf{q}$ in $\mathbb{R}^3$, the hemisphere $Hemi_{\mathbf{q}}$ is the set $\{\nu \in \mathbb{S}^2 \mid \langle \mathbf{q}, \nu \rangle \geq 0\}$, and where the support $supp(.)$ of a function is the closure of the set on which this function does not vanish. In other words, Lions et al. assume that there are no self-shadows. Also, such an assumption simplifies strongly the problem because we have then $R_L(\nu) = 0$ outside of $\mathbf{C}_{\mathcal{S},p}$ and so $\int_{\mathbf{C}_{\mathcal{S},p}} \langle \nu, \nu_p \rangle \, R_L(\nu) \, d\nu = \int_{\mathbb{S}^2} \langle \nu, \nu_p \rangle \, R_L(\nu) \, d\nu$ which completely removes the global dependency of the radiance with respects to the whole shape.

3.2 Formulation as a Partial Differential Equation

With the orthographic camera model, the image formation model above can be interpreted as a Partial Differential Equation (PDE) in the unknown function u:

$$I(x) = \int_{\mathbf{C}_{u,(x,u(x))}} \left\langle \frac{1}{\sqrt{1+|\nabla u(x)|^2}}(-\nabla u(x), 1)\ ,\ \nu \right\rangle^+ d\nu, \tag{4}$$

where $\mathbf{C}_{u,p}$ denote $\mathbf{C}_{\mathcal{S},p}$ (the surface $\mathcal{S}$ is represented by the function u). Solving the SFAS problem then amounts to integrating the PDE (4) given an image I. Clearly the result would be meaningful only if a solution exists, and if it is unique, or at least if one can characterize the set of functions u that are indistinguishable in the sense of all solving (4) for a given measured image I. Note that this equation is a first-order stationary **global integro-partial differential** equation of the general form: $H(x, u(x), \nabla u(x), u(.)) = 0, \quad \forall x \in Int(\mathfrak{D})$. The numerical and theoretical study of the solutions of these kind of equation is done via the *Hamiltonian*

$$H(x,t,p,u) = \int_{\mathbf{C}_{u,(x,t)}} \langle \frac{1}{\sqrt{1+|p|^2}}(-p,1), \nu\rangle^+ d\nu - I(x).$$

4 Analysis of the Shape From Ambient Shading Equation

In this section we consider the problem of uniqueness of solution of (4). While we show that the solution is, in general, not unique, we give an analytical characterization of all the different scenes that – under the given assumptions – yield the same measured image. This analysis is important both for the purpose of implementing viable numerical integration scheme, and also to make SFAS a useful tool in Computer Vision. This is akin to what is done in Structure From Motion [5], where the 3-D structure of a scene is in general not unique, but one can easily characterize the solutions as being equivalence classes under the similarity, affine or projective groups depending on knowledge on the camera calibration [6, 5, 13].

4.1 An Intrinsic Ambiguity

First, recall that $0 \leq R_{\mathcal{S}}(p) \leq \pi,\ p \in \mathcal{S}$ and $\mathbf{C}_{\mathcal{S},p} \subset Hemi_{\nu_p}$, so one can easily show that $R_{\mathcal{S}}(p) = \pi$ iff $\mathbf{C}_{\mathcal{S},p} = Hemi_{\nu_p}$. Now, let us consider a completely white image with a maximal intensity: $I(x) = \pi\ \forall\ x \in \mathfrak{D}$. With such an image, the solutions of equation (3) satisfy $\mathbf{C}_{\mathcal{S},p} = Hemi_{\nu_p}$ for all the points p on the surface. Therefore, if we represent the surface as the graph of the function u, it is easy to see that the surface lies below the tangent plane to the surface at the point $(x, u(x))$. So, the solutions u of (4) are concave, and so is the surface $\mathcal{S}$. Since inversely all concave functions generate such a white image then we can conclude that the set of solutions is comprised of all concave functions. In this case, the problem is clearly ill-posed because the image can be generated by a number of different surfaces, and therefore the solution cannot be unique. This problem does not arise only in this pathological case: It is patent as soon as the image contains a subset of pixels having the maximal intensity, as we illustrate in Figure 1. Pixels with maximal intensity are shown in red, and the green curve corresponds with a maximal solution when the blue gives the minimal one. Any curve between these two, which is concave on the set of points with maximal intensity, generates the same image as the one generated by the black curve. In the following sections, we will show that this condition is minimal, in the sense that the solution is unique if and only if there are no subsets of pixels having the maximal intensity. Also, when there are multiple solutions, they are characterized by in terms of their value on these subsets.

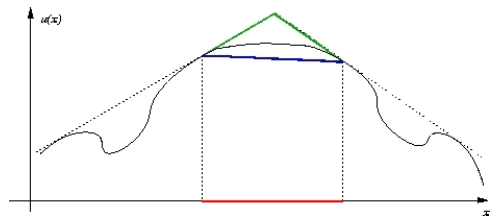


Fig. 1. Example of multiple solutions in dimension 2 when the image contains a subset of pixels having the maximal intensity. Any curve between the blue and the green curves, and which is concave on the set of points with maximal intensity, generates the same image as the one generated by the initial black curve.

4.2 Uniqueness Result and Characterization of the solutions

In this section we show that the solutions of the SFAS problem are charaterized by their value on the subset $\{x \mid I(x) = \pi\} \subset \mathfrak{D}$. To the end, let us define $\Omega = \{x \mid I(x) < \pi\}$ and let us complete the equation

$$H(x, u(x), \nabla u(x), u) = 0, \forall x \in \mathfrak{D} \tag{5}$$

by some Dirichlet boundary conditions on $\mathcal{C}\Omega = \mathfrak{D} - \Omega = \{x \in \mathfrak{D} \mid I(x) = \pi\}$. In other words, we assume that we know the height of the solution on this subset. The equation then becomes

$$\begin{cases} H(x, u(x), \nabla u(x), u) = 0, & \forall x \in \Omega, \\ u(x) = \varphi(x) & \forall x \in \mathcal{C}\Omega. \end{cases} \tag{6}$$

For mathematical convenience, we also assume that the brightness image I is continuous (then Ω is an open subset of $\mathfrak{D}$) and that the intensity is maximal on the boundary of the image (in other words, we assume that $\bar{\Omega} \subset \mathcal{I}nt\, \mathfrak{D}$). We can now state the uniqueness theorem:

Theorem 1. *If u and v are two C^1 solutions to equation* (6) *then $u = v$ on $\mathfrak{D}$.*

This theorem ensures that there exists at most a unique C^1 solution to equation (6). Also, it provides a characterization of the set of the solutions of equation (5), characterized by its values on the subset $\mathcal{C}\Omega$ (the region where $I(x) = \pi$). If the image never saturates ($\mathcal{C}\Omega$ is empty), then the solution is unique when complemented by a Dirichlet boundary condition. Equivalently, all solutions are parameterized by their boundary conditions. Because of space constraints, we cannot report the complete proof of theorem 1 here, and we refer the reader to our technical report [17] for details.

The relevance of this result from the standpoint of Computer Vision is that if we know the depth of the scene on the subset where the image is saturated, then there exists a unique solution to the Shape From Ambient Shading problem. This means that, elsewhere on the image, ambient shading is sufficient to recover the original surface which generated the image. In the next section we develop an approximation scheme for numerically integrating (6).

5 Approximation Scheme And Numerical Algorithm

In section 3 we have formalized the SFAS problem as the solution of a partial differential equation of the form $H(x, u(x), \nabla u(x), u) = 0$. We have then added Dirichlet boundary conditions on $\mathcal{C}\Omega = \mathfrak{D} - \Omega$ to arrive at a unique solution when the image is not saturated. In order to compute a reliable numerical solution to this equation, we use machinery available for Hamilton-Jacobi equations. The key point consists then in designing approximation schemes which are monotone [2, 1].

5.1 A Monotonic Scheme

An approximation scheme is a functional equation of the form

$$T(h, x, u^\rho) = 0 \qquad \forall x \in \bar{\Omega};$$

where T is a real function defined on $\mathbb{R}^+ \times \bar{\Omega} \times B(\bar{\Omega})$; $h \in \mathbb{R}^+$ defines the size of the grid that is used in the corresponding numerical algorithms (a 2D Cartesian grid); $B(\bar{\Omega})$ is the space of bounded functions defined on the set $\bar{\Omega}$. u^ρ is the unknown (u^ρ is a function). Also, we are interested in the solution u^ρ of the scheme T. Generally, we say that a scheme is monotone if for all $h \in \mathbb{R}^+$ and $x \in \bar{\Omega}$, the function $T(h, x, \cdot) : B(\bar{\Omega}) \to \mathbb{R}$ is monotone. That is, for all $y \in \bar{\Omega}$, $u(y) \geq v(y)$, then $T(h, x, u) \geq T(h, x, v)$. Following [1], we introduce the representation S of a scheme T as

$$S(h, x, u^\rho(x), u^\rho) = 0 \qquad \forall x \in \bar{\Omega}, \tag{7}$$

where $S : \mathbb{R}^+ \times \bar{\Omega} \times \mathbb{R} \times B(\bar{\Omega}) \to \mathbb{R} : (h, x, t, u) \mapsto S(h, x, t, u)$. Note that a representation of a scheme is also a scheme. This last mathematical object allows us to take advantage of the tools developed by Barles and Souganidis [1] which require that the scheme be monotonic with respect to all the values of u up to the value at one point, generally $u(x)$. We then isolate $u(x)$ from the other values of u. Also, let us stress that the result demonstrated by Barles and Souganidis [1] is optimal for a large class of (static as well as evolutive) Hamilton-Jacobi equations.

The introduction of representations is also a way to simplify the computations. In effect, the representation of a scheme $T(h, x, u^\rho) = 0$ by a scheme of the form $S(h, x, u^\rho(x), u^\rho) = 0$ suggests an iterative algorithm for computing a numerical approximation of the solution of the scheme. Given u^n (the approximation of u^ρ at step n), and a point x of $\bar{\Omega}$, the associated algorithm consists in solving the equation

$$S(h, x, t, u^n) = 0 \tag{8}$$

with respect to t. A solution of (8) is the updated value of u^n at x. Here, we are then going to use the definition of monotonicity given by Barles and Souganidis in [1]:

Definition 3 (monotonicity). *The scheme $S(h, x, u^\rho(x), u^\rho) = 0$ defined in $\bar{\Omega}$, is monotone if $\forall h \in \mathbb{R}^+, \forall x \in \bar{\Omega}, \forall t \in \mathbb{R}$ and $\forall u, v \in B(\bar{\Omega})$,*

$$u \leq v \quad \Longrightarrow \quad S(h, x, t, u) \geq S(h, x, t, v)$$

(the scheme is non-increasing with respect to u).

The interest of the monotonicity is twofold. (i) With other basic assumptions (monotonicity with respect to t, existence of a subsolution, bound for the subsolutions), this property is the key to ensure that the scheme is stable (existence of the solution and of an upper bound), that the computed approximations converge towards the solution of the scheme, see [15]. (ii) Combined with some stability and consistency properties, the monotonicity ensures that the solutions of the scheme converge towards the continuous solution of the considered PDE when the grid vanishes see [1]. In what follows, we are going to design a monotonic approximation scheme for the SFAS problem in order to take advantage of all these benefits.

5.2 Monotonic Scheme for the SFAS problem

For readability, we denote $H_{u,t}(x, p) = H(x, t, p, u)$. Let us recall that the Hamiltonian of insterest in SFAS is

$$H_{u,t}(x, p) = \int_{\mathbf{C}_{u,(x,t)}} \langle \frac{1}{\sqrt{1 + |p|^2}}(-p, 1), \nu \rangle^+ \, d\nu - I(x).$$

One can verify easily that $\mathbf{C}_{u,(x,t)}$ is decreasing (in the sense of inclusion) with respect to u and increasing with respect to t. Also, it follows that $H_{u,t}$ verifies exactly the same monotonic properties.

On the other hand, in order to get a consistent approximation scheme, we have to replace ∇u (represented by the variable p in the above Hamiltonian) in the PDE by one of its numerical approximations (finite differences). The difficulty is then to find such a discretization while maintaining monotonicity. In order to get a monotonic scheme, we take inspiration from *Lax-Friedrichs scheme* for conservation laws [3, 2]. We chose:

$$S(h, x, t, u) = H_{u,t}(x, \mathbf{D}u(x)) - \theta\, \mathbf{L}u^t(x), \tag{9}$$

where $\mathbf{D}u(x)$ is the vector obtained by a centered discretization of $\nabla u(x)$, more precisely, the ith component of $\mathbf{D}u(x)$ is

$$[\mathbf{D}u(x)]_i = \frac{u(x + h\overrightarrow{e_i}) - u(x - h\overrightarrow{e_i})}{2\,h}$$

and where $\mathbf{L}u^t(x)$ is the classical discretization of the Laplacian $\Delta u(x)$ (in which one replaces $u(x)$ by t), i.e.

$$\mathbf{L}u^t(x) = \sum_{i=1..N} \frac{u(x + h\overrightarrow{e_i}) + u(x - h\overrightarrow{e_i}) - 2t}{h^2}.$$

This scheme, however, is still not necessarily monotonic. To satisfy this property, we need to find an adequate value for θ. By differential calculus, one can verify that a sufficient condition to ensure this property is $\max_{i=1..N}\ h\,|\partial_{p_i} H_{u,t}(x, Dz)| \ \leq 2\theta$; see [17] for a detailled proof. By the same tools, one can also easily prove that $|\partial_{p_i} H_{u,t}(x, p)| \leq 2\sqrt{2}\pi$. The scheme(9) is then monotonic as soon as

$$\theta \geq \sqrt{2}\pi h.$$

Also, to limit the smoothing due to the Laplacian term introduced in the scheme (term which can be interpreted as a regularization), θ must be as small as possible.

On the other hand, under the assumptions of section 4.2, one can verify that any deep enough function is a subsolution of the scheme (9) (because the visibility cone becomes arbitrarily small). Moreover, the subsolutions are necessarily bounded by the function corresponding to convex hull defined by the Dirichlet boundary constraints. Since the scheme is also increasing with respect to t and verifies $\lim_{t\to+\infty} S(h, x, t, u) \geq 0$ then theorems 3.1 and 3.5 of [15] ensure that the scheme (9) is stable and that the iterative approximations converge towards the solution of the scheme.

In practice, we can start from any subsolution and we have just to update the surface with scheme (9) until convergence. Finally, our scheme being also consistent with the SFAS I-PDE, relying on Barle and Souganidis theorem [1], we can conjecture that the computed approximations converge towards the continuous solution of the I-PDE. This guarantees the reliability of our numerical approximations toward the theoretical solution of our problem.

6 Numerical Experiments

We focus here on the numerical results obtained by the algorithm associated to the scheme (9). As described in section 5.1, approximation schemes of the form (7) suggest an iterative numerical algorithm, whose udating step (at point x) consists in solving equation $S(h, x, t, u) = 0$ (equation in t), where u is the approximation of the whole solution at the previous step. Here, to solve

equation $H_{u,t}(x, \mathbf{D}u(x)) - \theta\ \mathbf{L}u^t(x) = 0$, we rewrite this equation as a fixed point equation $t = g(t)$, where

$$g(t) = \frac{1}{4}\left(\sum_{i=1,2}(u(x + h\overrightarrow{e_i}) + u(x - h\overrightarrow{e_i})) - \frac{h^2}{\theta}H_{u,t}(x, \mathbf{D}u(x)\right)$$

and then process the iterations $t_{n+1} = g(t_n)$. In practice this process systematically converges after less than 5 iterations (we assign t_0 to the previous value of $u(x)$). The numerical algorithm starts with a subsolution as a very steep valley such that visibility is closed to 0 for all points in the domain of the image. We refer the reader to [17] for further implemention details.

To test our algorithm, we consider some scenarios for which the problem is well-posed. In other words, we limit the computation domain to a subset of $\Omega = \{x \mid I(x) < \pi\}$. This computation domain is delimited by the red box in the corresponding figures. On the other part of the image domain, we enforce Dirichlet boundary conditions.

In our tests, we use the $\sin(x)\sin(y)$ surface. For the first test, we restrict the computation domain to a subset on which the surface is convex. As shown in Figure 2, the computed iterative solution converges accurately towards the original surface. In the second test, we want to extend

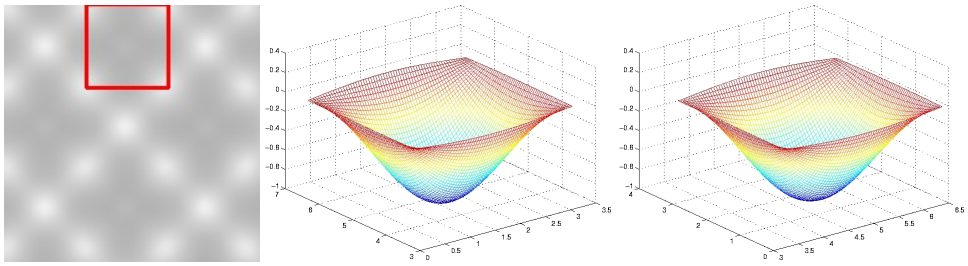


Fig. 2. Left: image generated by the $\sin x * \sin y$ surface with $h = 0.05$ and region of interest where we run the algorithm; middle: original surface (groundtruth) on the region interest; right: surface reconstructed by our algorithm (result).

the computation domain to both concave and convex areas. To remove the ambiguity due to points with maximal intensity, we reduce the intensity of the image by placing the $\sin(x)\sin(y)$ surface in a box, i.e. surrounded by four walls of a cube with the roof open. In this test, the algorithm converges towards the solution in both concave and convex regions. Nevertheless, as shown Figure 3, when the reconstruction is very accurate in the convex region, there is a significant error in the concave region.

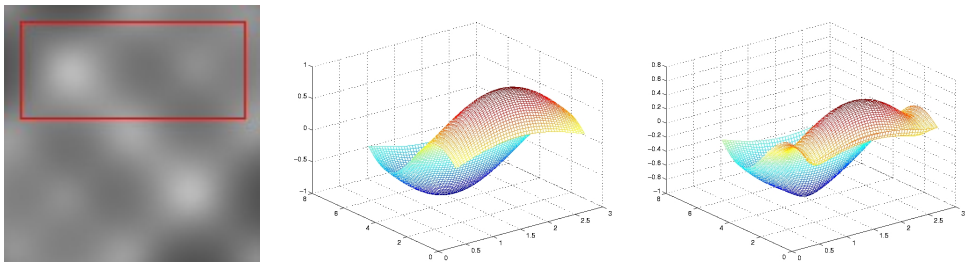


Fig. 3. Left: image generated by the $\sin x * \sin y$ surface with $h = 0.05$ inside a cubical box and region of interest where we run the algorithm; middle: original surface (groundtruth) on the region of interest; right: surface reconstructed by our algorithm (result).

Table 1 shows the minimum and maximum values of the original surfaces in the regions of interest (where the algorithm is applied). It also shows the L_1, L_2 and L_∞ errors. The top row shows the errors for the first test ($\sin(x)\sin(y)$ surface) illustrated in Figure 2. The second row shows the errors for $\sin(x)\sin(y)$ surface inside a box; it corresponds with the result of Figure 3. In our experiments, we have used the L_1 error to test for convergence.

In the second test, one can understand the error on the concave region as a result of the introduction of the regularization term (which was needed to make the scheme monotonic). To further analyze this effect, we focus on the concave part and we perform the following two experiments. 1) We run our algorithm with an input image containing the regularization term. More precisely, we use

$$\tilde{I}(x) = \int_{\mathbf{C}_{u,(x,u(x))}} \langle \frac{1}{\sqrt{1+|\mathbf{D}u(x)|^2}}(-\mathbf{D}u(x),1),\nu\rangle^+ \, d\nu - \theta\, \mathbf{L}u(x)$$

as input to our algorithm. So, in practice, the algorithm computes the solution of equation $\int_{\mathbf{C}_{u,(x,u(x))}} \langle \frac{1}{\sqrt{1+|\mathbf{D}u(x)|^2}}(-\mathbf{D}u(x),1),\nu\rangle^+ \, d\nu - \tilde{I}(x) - \theta\, \mathbf{L}u(x) = 0$ and the computed solution should then better coincide with the original surface. We then make this third test with the $\sin x \sin y$ surface inside the box (with a computation domain reduced to the concave part). As shown in table 2 and Figure 4, the algorithm is now able to recover accurately the surface. 2) Finally, since the regularization parameter θ is linearly dependent with the size of the grid h, then the regularization effect should reduce when the size of the grid vanishes. We then redo the second test ($\sin x \sin y$ surface inside a box, with the original image I, with the same reduced computation domain as previously) with smaller and smaller grid sizes: $h = 0.2, 0.1, 0.08, 0.05, 0.04$. Also, as we can see in Figure 5 and Table 3, the computed approximations actually converge towards the original surface when the grid size is reduced. In addition to confirm the above assertion, this also validates our methodology and our theory which ensures a well-posed algorithm whose the output convergences towards the continuous solution when the grid vanishes.

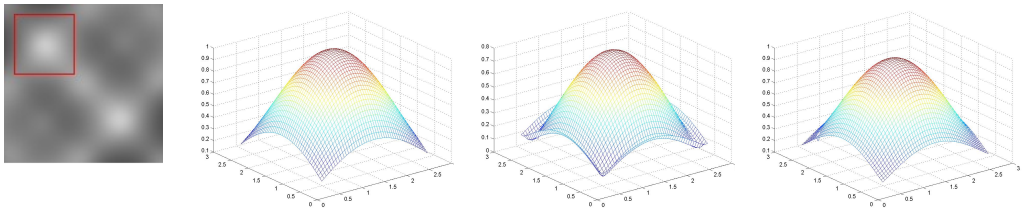


Fig. 4. $sinx * siny$ image with regularization and region of interest where we run the numerical scheme. Results of the numerical scheme with (right) and without (left) regularization in input image.

Table 1. Errors for the first two tests.

	min value	max value	L_1 errors	L_2 errors	L_∞ errors
$\sin x \sin y$, Fig. 2	-0.999707	0.066750	0.006191	0.009792	0.033867
$\sin x \sin y$ in box, Fig. 3	-0.999707	0.999568	0.188896	0.240712	0.372564

Table 2. Errors by adding the regularization term in the input image.

	Min Value	Max Value	L_1 Error	L_2 Error	L_∞ Error
without regularization	-0.999707	0.999568	0.186037	0.189434	0.207331
with regularization	-0.999707	0.999568	0.065627	0.067900	0.078941

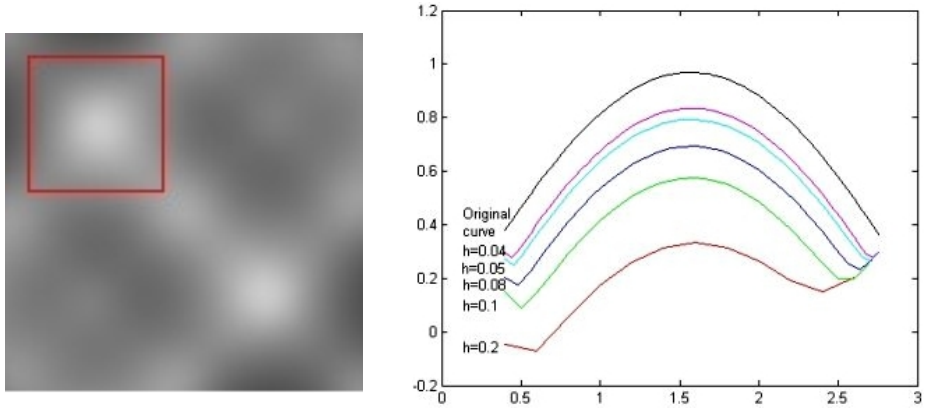


Fig. 5. Reconstruction with different grid sizes h.

Table 3. Errors with respect to h.

grid sizes (h)	$h = 0.2$	$h = 0.1$	$h = 0.08$	$h = 0.05$	$h = 0.04$
L_1 error	0.504147	0.358676	0.270054	0.186037	0.151427
L_2 error	0.526644	0.371685	0.276862	0.189434	0.153691
L_∞ error	0.658852	0.424875	0.308127	0.207331	0.166671

7 Conclusion and Future Work

In 3-D reconstruction approaches to Computer Vision, illumination is rarely modeled explicitly. With few notable exceptions, most work in Structure From Motion assumes that illumination is constant and therefore it ascribes all photometric effects to the radiance of the scene, regardless of how it comes to be. In Shape From Shading, where the illumination is key, most existing work models it as an ideal point light source. In this paper we focus on the opposite abstraction, where the illumination is diffuse, and indeed it is constant. Outdoor scenes on a cloudy day, or indoor scenes in modern offices are reasonably well approximated by these conditions. Clearly one would like to account for arbitrary unknown radiant distributions, and possibly also illumination, but this would render the analysis prohibitive.

Already under the restrictive assumptions we have chosen to operate under, the problem of recovering the 3-D shape of the scene translates to a global integro-differential equation that, to the best of our knowledge, has never been analyzed. Although algorithms have been explored in the past to exploit diffuse shading for recovering properties of the scene, a thorough theoretical study of the mathematical properties of this problem has been lacking.

We believe we are the first to study the uniqueness of SFAS, to show that – in general – it is not unique, and to characterize the set of scenes that are indistinguishable, in the sense of satisfying the assumptions of SFAS and generating the same image.

While we believe that the main contribution of this paper is analytical, we do validate our results empirically in simulation. To that end, we propose a monotonic scheme for numerically integrating the SFAS equation, and show experimental results that highlight the features, and challenges, of this method.

Moving forward it would be desirable to develop both numerical schemes that are robust to noise, and – most importantly – algorithms that can provide competing explanations for the

image so that different assumptions, among which those for SFAS, can be applied to different portions of the image. For instance, it would not make sense to apply SFAS in regions of the image with albedo discontinuities (where stereo works well), and vice-versa one would not want to use stereo or Structure From Motion where albedo is constant. It would also be desirable to integrate SFAS with multiple-view reconstruction by providing constraints on multiple images of the same scene seen from multiple viewpoints. Finally, it would be desirable to relax the assumptions of pinhole orthographic camera to a perspective camera with a finite aperture.

Acknowledgement: Research was supported by the Flamenco project ANR-06-MDCA-007 and ONR N00014-08-1-0414.

References

1. G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis*, 4:271–283, 1991.
2. M. G. Crandall and P. L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, 43(167):1–19, 1984.
3. M. G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Mathematics of Computation*, 34(149):1–21, January 1980.
4. J.-D. Durou, M. Falcone, and M. Sagona. Numerical methods for shape-from-shading: A new survey with benchmarks. *Computer Vision and Image Understanding*, 109(1):22–43, jan 2008.
5. O.D. Faugeras. Three-dimensional computer vision: A geometric viewpoint. *MIT Press*, 1993.
6. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
7. B.K. Horn. *Robot Vision*. mit-press, 1986.
8. B.K. Horn and M.J. Brooks, editors. *Shape from Shading*. The MIT Press, 1989.
9. J.J. Koenderink, S. C. Pont, A. J. van Doorn, A. M. L. Kappers, and J. T. Todd. The visual light field. *Perception*, 36:1595–1610, 2007.
10. M. S. Langer and H. H. Bulthoff. Depth discrimination from shading under diffuse lighting. *Perception*, 29(6):649–660, 2000.
11. M. S. Langer and S. W. Zucker. Shape from shading on a cloudy day. *Journal of Optical Society of America*, 11:467–478, 1994.
12. P.-L. Lions, E. Rouy, and A. Tourin. Shape-from-shading, viscosity solutions and edges. *Numer. Math.*, 64:323–353, 1993.
13. Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3D vision, from images to models*. Springer Verlag, 2003.
14. S. Nayar, K. Ikeuchi, and T. Kanade. Shape from interreflections. *IJCV*, 6(3):173–195, 1991.
15. E. Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem*. PhD thesis, Univ. of Nice-Sophia Antipolis, 2004.
16. E. Prados and O. Faugeras. Shape from shading: a well-posed problem ? In *Proceedings of CVPR'05*, volume II, pages 870–877. IEEE, jun 2005.
17. E. Prados, N. Jindal, and S. Soatto. A non-local approach to shape from ambient shading. Technical report, INRIA, 2009.
18. A.J. Stewart and M. S. Langer. Towards accurate recovery of shape from shading under diffuse lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1020–1025, 1997.
19. Y.L Tian, H.T. Tsui, S.Y. Yeung, and S. Ma. Shape from shading for multiple light sources. *Journal of the Optical Society of America*, 16(1):36–52, January 1999.
20. T. Wada, H. Ukida, and T. Matsuyama. Shape from shading with interreflections under proximal light source-3D shape reconstruction of unfolded book surface from a scanner image. In *ICCV*, 1995.
21. J. Yang, D. Zhang, N. Ohnishi, and N. Sugie. Determining a polyhedral shape using interreflections. In *CVPR'97*, page 110, 1997.
22. R. Zhang, P.-S. Tsai, J.-E. Cryer, and M. Shah. Shape from Shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.

Quel développement durable ?

Emmanuel Prados[1,2], Elise Arnaud[2,1]

avec la collaboration de
Francois Mancebo[3,1], Pierre-Yves Longaretti[4,1]
et Dominique Chouchan[5]

juillet 2010

Résumé

Si tout le monde, ou presque, s'accorde sur la nécessité d'un développement durable, une certaine cacophonie règne dès qu'il s'agit de rendre des arbitrages sur des solutions concrètes. Des outils d'aide à la décision pourraient faciliter la tâche des acteurs institutionnels.

Le concept de développement durable intègre trois dimensions fondamentales : l'environnement bien sûr, mais aussi le développement social et le développement économique. Pour nous, mathématiciens et informaticiens, travailler sur le développement durable nous conduit donc d'emblée à adopter une approche systémique, afin de pouvoir prendre en compte ces trois dimensions. C'est le défi que notre équipe s'est lancé en s'engageant dans une démarche de modélisation de ces systèmes complexes. Notre but est de mettre au point des outils d'aide à la décision destinés aux acteurs politiques pour anticiper plus rationnellement qu'aujourd'hui les effets d'une décision.

Jusqu'à ce jour, les politiques de développement durable ont essentiellement porté sur des questions d'environnement et d'énergie. Plusieurs raisons à cela. D'abord, on commence à bien connaître les risques en la matière. Ensuite, la prise de conscience par la population est encore insuffisante sur les autres aspects, outre le fait que les connaissances scientifiques manquent cruellement et que la définition même du développement durable et ce qu'elle recouvre exactement restent encore flous [5]. Or un décideur peut souhaiter apprécier l'impact social d'une mesure environnementale, par exemple, ne serait-ce que pour s'assurer qu'elle ne risque pas de renforcer des inégalités. C'est ainsi que nous en sommes arrivés à la notion de « fausses bonnes idées » dont nous pensons que l'émergence découle pour une part de l'absence d'approche globale. Du coup, des mesures d'apparence raisonnable peuvent se traduire par des effets pervers qui annihilent les effets bénéfiques attendus. En travaillant sur une facette on

1. Equipe STEEP, INRIA Rhône-Alpes
2. Laboratoire Jean Kuntzmann (LJK)
3. Laboratoire PACTE de Grenoble et Institut d'Aménagement des Territoires, d'Environnement et d'Urbanisme de l'Université de Reims
4. LAOG, OSUG, Grenoble
5. Journaliste scientifique

peut en dégrader une autre : par exemple l'économie si on ne travaille que sur l'environnement, ou le social si on ne considère que l'économie et l'environnement. Exemple : la politique de traitement des déchets mise en oeuvre dans le canton de Vaud, en Suisse (voir l'encadré).

Une autre difficulté vient de ce qu'il est d'usage d'appeler les "effets rebonds" : ces effets sont liés à des boucles de rétroaction complexes et mal identifiées consécutives à une action politique *a priori* bonne, mais qui se trouve ainsi pervertie. Un exemple emblématique est celui des ampoules à faible consommation d'énergie. Non seulement elles contiennent du mercure, extrêmement toxique, mais les consommateurs ont cru qu'elles suffiraient à réduire leur consommation globale d'électricité. En fait, en toute bonne conscience, ils ont augmenté leurs puissance et durée d'éclairage, mais sans modifier d'un iota leur mode de consommation d'énergie de manière générale.

Une bonne idée pervertie

Comment le contexte culturel et socio-économique peut rendre une bonne idée infructueuse ? Il y a une dizaine d'années, le canton de Vaud (Suisse) s'est engagé dans une politique de valorisation des déchets (tri sélectif, recyclage, etc.) afin de réduire les pollutions environnementales dues aux incinérateurs. Une taxe de type pollueur-payeur a été mise en place de manière à financer les investissements réalisés pour les déchets domestiques. Dans un premier temps, cette taxe a été calculée sur le volume des déchets. Du coup, les citoyens les ont compressés. Deuxième étape : la taxe a été établie sur la base du poids de ces déchets,... les ménages ont alors massivement brûlé leurs déchets dans leur jardin ! Résultat : les niveaux de dioxines et de furane mesurés quelques années plus tard dans le sol ont été plus élevés que ceux mesurés avant la mise en place de cette politique.

François Mancebo, université de Reims, laboratoire Pacte, membre de l'équipe STEEP

Ces quelques remarques mettent en lumière un point décisif à nos yeux : la question de l'arbitrage. Car non seulement les problèmes doivent être abordés dans le cadre d'une démarche systémique, mais il n'y a aucune raison, aucune loi établie, qui permettrait de postuler que des objectifs purement environnementaux soient compatibles avec des objectifs de progrès social ou bien de croissance économique. Aux politiques donc la lourde tâche de trouver un équilibre entre ces différentes dimensions, et de déterminer quel niveau de dégradation ils sont prêts à concéder et à faire admettre par la société civile sur l'une ou l'autre de ces dimensions ou sur les trois.

Les outils numériques d'aide à la décision que nous allons développer visent précisément à leur donner des moyens pour anticiper et évaluer les conséquences de tel ou tel choix, par exemple le choix d'une taxe carbone, de l'installation d'un parc éolien, etc. De surcroît, il serait contre-productif de passer d'un état à un autre sans élaborer des politiques de transition : on imagine mal le passage brutal d'un état où la circulation automobile est autorisée en ville à un état où seuls les vélos pourraient rouler. Cela suppose donc de faire en sorte que ces

transitions soient elles aussi modélisables par nos outils.

Quels types d'outils mettre au point pour analyser et visualiser des systèmes aussi complexes, tout en anticipant les rétroactions et en comprenant les dynamiques en jeu ? On en distingue deux grandes catégories : ceux fondés sur des techniques de simulation et ceux s'appuyant sur des méthodes d'optimisation. Les premiers visent à prédire le comportement d'un tel système en réaction à des modifications de paramètres géophysiques (réchauffement climatique...), biologiques (réduction de la biodiversité...), économiques (crise financière...), etc. et aux choix politiques envisagés. Ce sont des outils de prospective. Par exemple, avant la construction d'un axe périphérique autour d'une ville, un outil de simulation permettrait d'estimer les impacts économiques, sociaux et environnementaux engendrés par les rétroactions liées à l'amplification de l'étalement urbain.

Quant aux méthodes d'optimisation, leur but est d'améliorer l'« efficacité » des choix. Elles permettent en effet de déterminer par des méthodes numériques la suite de décisions propre à minimiser un coût ; qu'il soit économique, environnemental ou social. Supposons par exemple qu'une collectivité locale veuille prendre des mesures pour respecter ses engagements en matière d'émissions de CO_2, mais de telle sorte que ces mesures soient acceptables par la société civile (notamment en termes de dépenses par foyer). L'idée est d'évaluer les mesures incitatives nécessaires (pour des travaux d'isolation ou de modernisation des systèmes de chauffage, en faveur des transports en commun, etc.) qui soient à la fois efficaces et de coût minimal. Dans ce cas, la fonction à minimiser est le coût, et les contraintes sont les émissions de CO_2, la rationalité et l'acceptabilité.

Notre priorité est l'échelle locale, du bassin d'emploi d'une ville à la région, l'échelle régionale nous semblant la plus pertinente. C'est en effet à cette échelle que l'on peut espérer la transition la plus rapide vers un développement durable, sachant que les ressources disponibles diffèrent d'un territoire à l'autre, de même que les comportements culturels : une politique de développement durable ne peut être identique en Rhône- Alpes et en Bretagne. D'autres facteurs nous confortent dans ce choix : l'indépendance croissante des régions ainsi que leur plus grande réactivité et marge de manoeuvre par rapport aux politiques nationales. La région est également l'unité territoriale à laquelle se fera une relocalisation partielle des échanges économiques si besoin est, notamment dans les domaines agricole et énergétique. Enfin, alors que les phénomènes au plan mondial ont fait l'objet de nombreuses études [6, 3], peu de travaux portent sur le niveau régional.

Or dans les prochaines années, les décideurs locaux vont devoir eux aussi prendre des décisions pour respecter les engagements par exemple pris à Kyoto, lors du Grenelle de l'environnement, etc. Il y a donc urgence. Pour y répondre, nous avons choisi de nous appuyer dans un premier temps sur des modèles déjà opérationnels. Plus précisément, nous allons combiner trois modèles existants [8, 7, 2] : un modèle couplant le transport et l'usage des sols [4] (voir la figure 1), un modèle couplant énergie, climat et qualité de l'air, et un modèle permettant de modéliser les services des écosystèmes (pollinisation, stabilité des sols, protection contre les inondations, régulation du climat...). L'idée est de proposer un outil permettant de simuler les principaux leviers d'actions des politiques locales (aménagement et urbanisme, transports, bâtiment) tout en intégrant les impacts sociaux et environnementaux.

Ce programme de travail ne va évidemment pas sans difficultés. Première

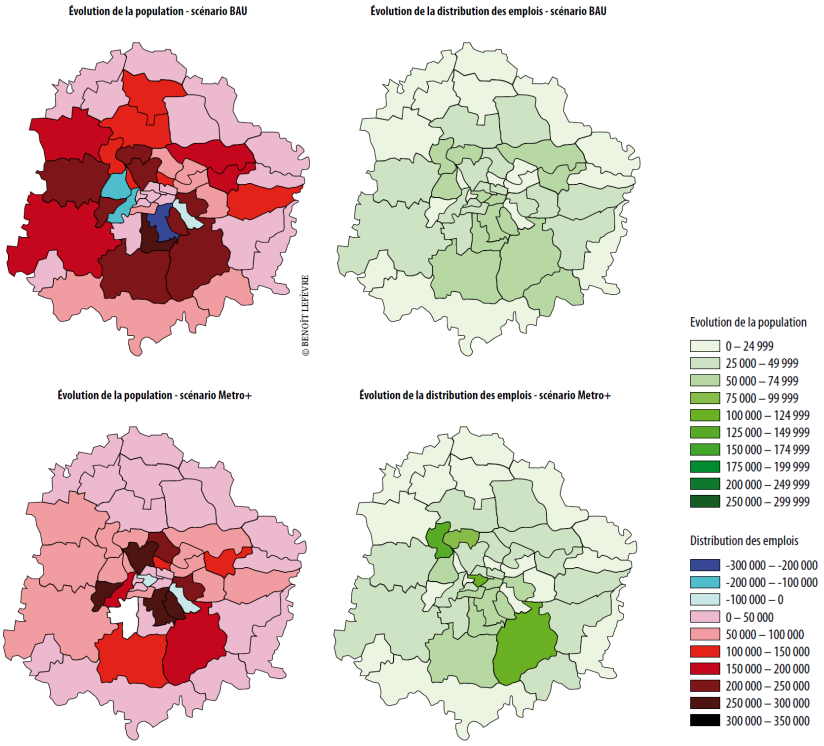


FIGURE 1 – **Ces images montrent le résultat de simulations réalisées par Benoît Lefèvre [4] sur la base d'un modèle couplant transport et usage des sols pour la ville de Bangalore (Inde). En (a) et (b) : les évolutions de la répartition spatiale de la population et de l'emploi d'ici à 20 ans sans aucune politique particulière (Business As Usual, ou BAU). La population part vers la périphérie de la ville, la localisation des nouveaux emplois restant homogène. En (c) et (d) : les évolutions des mêmes paramètres, mais selon un scénario baptisé METRO+, c'est-à-dire avec la mise en place de deux lignes de métro (nord/sud et est/ouest) et d'une politique d'aménagement urbain. On observe alors une concentration des emplois dans la première couronne et au sud de la deuxième. La population se concentre dans les zones d'emploi et dans la première couronne, où le prix du foncier est moins élevé qu'au centre ville mais qui est desservie par le métro.**

difficulté : il faut choisir le degré de finesse de la modélisation. D'un côté, il est essentiel de limiter le niveau de complexité des modèles, afin de garantir des temps de calcul raisonnables et la convergence des algorithmes. De l'autre, les modèles doivent être assez fins et détaillés pour être pertinents et réalistes. Il faut également faire des choix sur ce qui doit rester de l'ordre de l'hypothèse (scénarios a priori) et sur ce qui doit être modélisé. Par exemple, la demande énergétique peut être soit considérée comme une donnée initiale (scénario relatif à cette demande), soit modélisée de manière à rendre compte des rétroactions entre coût de l'énergie et demande.

Une seconde difficulté résulte du caractère multi-échelle des problèmes posés. Cet aspect découle de la prise en compte simultanée d'acteurs de nature différente et évoluant à des échelles spatiales et temporelles spécifiques. L'enjeu consiste à utiliser des représentations des variables et des données adaptées à l'hétérogénéité spatiale et temporelle des structures et des processus. Par exemple, l'échelle pour traiter des questions d'aménagement d'un quartier est différente de celle à laquelle nous modéliserons certains phénomènes biologiques et plus généralement les services des écosystèmes.

La troisième difficulté concerne la gestion des diverses sortes d'incertitudes. La quantification des incertitudes pour des problèmes de cette complexité est certainement l'un des plus grands défis que nous devons relever. Ces incertitudes apparaissent à plusieurs niveaux : elles vont de l'imprécision des données jusqu'aux incertitudes liées à l'absence de connaissance scientifique sur certains des processus concernés. Or les méthodes actuelles se limitent à donner des résultats prospectifs déterministes, sans évaluation de la confiance que l'on peut avoir en ces résultats. Notre équipe travaille en particulier sur l'analyse des trois types d'incertitudes les plus importants : celles liées au choix des scénarios, celles dues aux erreurs contenues dans les données et celles générées par le modèle.

Pour réaliser ce travail, nous nous sommes constitués en équipe pluridisciplinaire, avec notamment un urbaniste géographe et chercheur compétent en physique et géophysique. Nous allons en outre nous appuyer sur le réseau régional SOCLE^3 [1] qui regroupe toutes les compétences scientifiques nécessaires pour un tel projet : climatologie, sciences politiques, économie, énergie, biologie, sciences humaines et sociales... Notre objectif est bien sûr de concevoir des outils et des méthodologies les plus génériques possible. Mais dans un premier temps, nous testerons nos idées sur la zone du Schéma de cohérence territoriale (SCOT) de la région urbaine Grenobloise, complétée par les zones naturelles qui lui sont directement connectées, et sur la région Rhône-Alpes. Nous travaillerons en étroite collaboration avec les agences locales de l'énergie, de l'urbanisme etc. Notre modèle systémique devrait être opérationnel d'ici à trois ans.

Références

[1] SOCLE3 group, http ://socle3.obs.ujf-grenoble.fr/.

[2] Roelof Boumans, Robert Costanza, Joshua Farley, Matthew A. Wilson, Rosimeiry Portela, Jan Rotmans, Ferdinando Villa, and Monica Grasso. Modeling the dynamics of the integrated earth system and the value of global ecosystem services using the gumbo model. *Ecological Economics*, 41(3) :529–560, June 2002.

[3] Lester R. Brown. *Le plan B : Pour un pacte écologique mondial*. Calmann-Lévy, 2007.

[4] B. Lefevre. *La soutenabilité environnementale des transports urbains dans les villes du sud. Le couple "transport-usage des sols" au coeur des dynamiques urbaines*. Economie et finance, CERNA - Centre d'Economie Industrielle, ENSMP, december 2007.

[5] F. Mancebo. Le développement durable en questions. *Cybergeo, European Journal of Geography*, épistémologie, histoire, didactique(404), 2007.

[6] Donella H. Meadows, Jorgen Randers, and Dennis L. Meadows. *Limits to Growth : The 30-Year Update*. Chelsea Green, 2004.

[7] Ad J. Seebregts, Gary A. Goldstein, and Koen Smekens. Energy/environmental modelling with the markal family of models. Technical Report ECN-RX-01-039, Energy research Centre of the Netherlands (ECN), 2001.

[8] M. Wegener. *Handbook of transport geography and spatial systems*, volume 5 of *Handbooks in Transport*, chapter Overview of land use transport models. Elsevier Science Ltd, August 2004.

Troisième partie

Conclusion

Conclusion

Dans ce document, j'ai donné un aperçu de mes activités de recherche ainsi que de mes activités administratives ou en lien avec la recherche pour la période allant de 2006 à 2011. La description de mes résultats scientifiques est complétée par une collection de mes articles les plus représentatifs.

Pour moi, l'écriture de ce document me permet de tourner définitivement une page, celle de la vision par ordinateur ; domaine auquel je me serais totalement consacré pendant plus de six ans (en prenant en compte mes années de thèse et de post-doctorat). Malgré le plaisir intense que j'ai éprouvé à travailler dans ce domaine, depuis 2008, je me suis petit à petit tourné vers d'autres problématiques plus en phase avec les préoccupations sociétales actuelles. Désormais je peux me vouer totalement à mes nouvelles activités de recherche et à l'épanouissement de l'équipe STEEP.