



HAL
open science

Reconstruction d'un modèle géométrique à partir d'un maillage 3D issu d'un scanner surfacique

Silvère Gauthier

► **To cite this version:**

Silvère Gauthier. Reconstruction d'un modèle géométrique à partir d'un maillage 3D issu d'un scanner surfacique. Ingénierie assistée par ordinateur. Université Montpellier, 2018. Français. NNT : 2018MONT018 . tel-02076352

HAL Id: tel-02076352

<https://theses.hal.science/tel-02076352>

Submitted on 22 Mar 2019

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.

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En informatique

École doctorale I2S - Information, Structures, Systèmes

Unité de recherche LIRMM - Laboratoire d'Informatique, de Robotique et de Micro-électronique de Montpellier

Reconstruction d'un modèle géométrique à partir d'un maillage 3D issu d'un scanner surfacique

Présentée par **Silvère GAUTHIER**

Le 17 octobre 2018

Sous la direction de **William PUECH**
et **Gérard SUBSOL**

Devant le jury composé de

Géraldine MORIN, Prof. des Universités, ENSEEITH / IRIT Toulouse

Jean-Claude LEON, Prof. des Universités, INPG / INRIA Grenoble

Guillaume LAVOUE, MCF HDR, INSA / LIRIS Lyon

Gilles GESQUIERE, Prof. des Universités, Lyon 2 / LIRIS Lyon

William PUECH, Prof. des Universités, Univ. Montpellier / LIRMM Montpellier

Gérard SUBSOL, CR CNRS, LIRMM Montpellier

Roseline BENIERE, Responsable R&D, C4W Montpellier

Président du jury

Rapporteur

Rapporteur

Examineur

Directeur de thèse

Co-encadrant

Invité



UNIVERSITÉ
DE MONTPELLIER

Remerciements

On dit souvent d'une thèse que c'est un travail personnel. En réalité, chaque personne de l'entourage, professionnel comme personnel, participe à sa manière à l'élaboration de la thèse. Je vais essayer de tous vous remercier, en essayant de n'oublier personne.

Je voudrais tout d'abord remercier les membres de mon jury pour leurs remarques et questions très pertinentes, qui ont permis de faire ressortir des aspects de la thèse importants à mes yeux. Merci à Géraldine MORIN pour avoir examiné mon travail et présidé mon jury de thèse. Merci à Jean-Claude LEON et Guillaume LAVOUE pour avoir accepté d'être rapporteurs et d'avoir participé à l'amélioration du manuscrit mais aussi de la soutenance. Merci à Gilles GESQUIERE pour avoir pris le temps d'examiner mon travail.

Je voudrais aussi remercier mes trois encadrants qui m'ont permis d'en arriver là. Merci à Roseline BENIERE pour m'avoir accompagné dans l'entreprise depuis mes début en tant que stagiaire jusqu'à obtenir mon grade de docteur, tant pour le côté industriel que pour le côté scientifique. Le temps passé ensemble a pu rentabiliser l'open space et les salles de réunion de l'entreprise. Merci à Gérard SUBSOL pour m'avoir proposé de nombreuses pistes et avec qui j'ai pu discuter de nombreux points techniques de manière très fine. J'aimerais aussi remercier tes 17 stylos rouges pour avoir donné leur vie pour la bonne cause. Merci à William PUECH pour m'avoir suggéré de faire mon stage chez C4W, puis d'enchaîner sur la thèse, et pour m'avoir accompagné tout du long avec enthousiasme. J'espère pouvoir par la suite garder contact depuis l'entreprise que tu m'as également suggérée!

J'ai eu la chance de faire partie, pendant ces trois ans, de l'équipe ICAR du LIRMM. Je remercie tous les membres de l'équipe, permanents comme doctorants (y compris ceux qui sont déjà partis), pour tous les moments passés ensemble. Vous allez me manquer, ainsi que toutes nos discussions qui portaient dans tous les sens autour d'un café ou d'un thé, même si je n'en bois toujours pas. Les conversations survoltées avec mon co-locataire de bureau vont également me manquer. J'espère tout de même pouvoir garder le contact, et appliquer ce que j'ai appris : la cohésion d'équipe passe par une bonne machine à café, des croissants... et de l'humour!

J'ai aussi eu la chance de faire partie de l'équipe de C4W, ce qui a permis de valoriser mes travaux à travers des applications déjà utilisées par des clients, ce qui me tenait à coeur. Je remercie donc chaque membre de l'équipe, pour l'intégration dont j'ai eu droit, qu'elle soit logicielle ou sociale. Je n'oublierai pas les bons moments passés avec vous pendant les pauses déjeuner, y compris avec ceux qui ne sont plus de la partie. Merci également à Gildas POUESSEL et François LE BRETON pour avoir suivi et participé à mes travaux de thèse.

Je n'en serais pas là non plus si je n'avais pas pu décompresser et me reposer de temps en temps. Je remercie donc mes amis et ma famille pour m'avoir écouté parler de ma thèse sans rien n'y comprendre, et pour m'avoir soutenu jusqu'au bout. Je remercie en particulier ma femme Laëtitia, qui a du me supporter pendant plus de trois ans et sans qui je n'aurais probablement pas pu réussir. Merci de m'avoir soutenu.

Un grand merci à tous pour avoir participé à cette thèse!

Table des matières

Introduction	1
I Etat de l'art	7
1 Extraction de primitives géométriques à partir d'un maillage 3D	9
1.1 Introduction	10
1.2 Méthodes incrémentales	12
1.3 Méthodes directes	35
1.4 Synthèse	38
2 Analyse et embellissement d'un ensemble de primitives pour la rétro-ingénierie	39
2.1 Introduction	40
2.2 Reconnaissance d'un système de référence en CAO	43
2.3 Reconnaissance de motifs particuliers	46
2.4 Application à l'embellissement	56
2.5 Synthèse	59
II Contributions	61
3 Procédure robuste d'extraction de primitives dans des maillages 3D numérisés	63
3.1 Introduction	64
3.2 Segmentation "arêtes vives"	65
3.3 Ajustement global des primitives	76
3.4 Segmentation "arêtes lisses"	80
3.5 Adaptation automatique de l'ajustement local des primitives	82
3.6 Résultats expérimentaux	87
3.7 Conclusion	90
4 Analyse sémantique et embellissement du modèle reconstruit	91
4.1 Introduction	92
4.2 Construction d'un graphe de relations entre les primitives	93
4.3 Analyse globale et embellissement des primitives	98
4.4 Reconnaissance de motifs particuliers	111
4.5 Expérience d'embellissement intégré à la chaîne de traitement	123
4.6 Conclusion	128
Conclusion et perspectives	129
Liste des publications	135
Références	137



Introduction

Depuis quelques années, les systèmes de numérisation et de fabrication d'objets 3D sont facilement accessibles. D'une part, la numérisation d'un objet réel se fait généralement à l'aide d'un scanner 3D [92], et d'autre part, un objet peut être fabriqué facilement grâce aux technologies d'impression 3D [117].

Prenons l'exemple d'un appareil vendu en pièces détachées. A partir d'un certain nombre d'années, les pièces standards peuvent ne plus être fabriquées ou difficiles à trouver. Grâce aux technologies de numérisation et d'impression 3D, il devient alors possible, pour un particulier ou un industriel, de numériser et fabriquer ces pièces détachées au besoin et directement sur place. La fabrication d'une pièce nécessite au préalable un modèle 3D, obtenu soit avec un logiciel de CAO (Conception Assistée par Ordinateur), soit par numérisation 3D. Cependant, la numérisation 3D ne fournit pas le même type de modèle. En effet, un logiciel CAO permet de définir la géométrie et la topologie exacte d'un objet, à travers une représentation géométrique précise, alors que la numérisation 3D construit une représentation discrète, à partir de points 3D acquis sur la surface de l'objet réel. De plus, ces points ne fournissent pas directement d'information sur la géométrie de l'objet. Or, dans le cas où l'utilisateur veut modifier l'objet numérisé avant de le fabriquer à nouveau, par exemple en changeant le diamètre d'un perçage, il est nécessaire de pouvoir retrouver cette géométrie. Nous avons alors besoin d'algorithmes permettant de transformer un modèle d'une représentation à l'autre.

Ainsi, nous pouvons définir deux types de méthode. Pour passer d'une représentation continue à une représentation discrète, nous devons réaliser une "discrétisation". Celle-ci consiste essentiellement à échantillonner des points 3D sur les surfaces géométriques, définies par la représentation continue, afin de reconstruire un modèle discret. Pour cela, de nombreuses méthodes existent et cherchent à respecter au mieux la géométrie sous-jacente [25]. A l'inverse, pour passer d'une représentation continue à une représentation discrète, nous devons appliquer un processus de "rétro-ingénierie". Celui-ci a pour objectif d'extraire la géométrie de l'objet à partir des points 3D, puis de reconstruire le modèle continu [5].

Notons qu'au début du scénario que nous proposons, l'utilisateur n'a en sa possession que l'objet réel. En effet, le modèle numérique d'une pièce détachée n'est généralement pas fourni avec. Dans ce cas, le processus de modification de la pièce consiste à :

1. numériser la pièce à l'aide d'un scanner 3D, fournissant une représentation discrète ;
2. reconstruire une représentation continue à l'aide d'un processus de rétro-ingénierie ;
3. modifier la représentation continue à l'aide d'un logiciel de CAO ;
4. discrétiser et fabriquer la pièce modifiée avec une imprimante 3D.

Après avoir décrit les principales représentations continues et discrètes, nous nous concentrons sur la deuxième étape de ce scénario. Nous présentons ensuite le positionnement de la thèse et les travaux réalisés, qui s'inscrivent dans un processus de rétro-ingénierie à partir de pièces numérisées.

Représentation continue

Il existe deux principaux types de représentation continue (Fig. 0.1). La première, appelée “CSG”, permet de construire un objet par une combinaison de volumes [16]. La seconde, appelée “B-Rep”, définit une combinaison de surfaces [101].

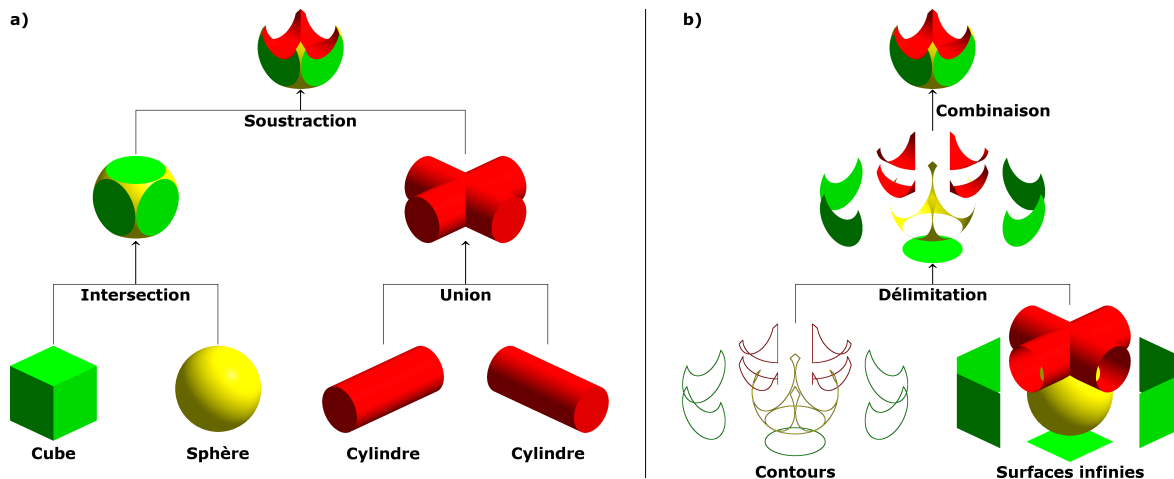


FIGURE 0.1 – Deux représentations continues du même objet : a) Modèle CSG, b) Modèle B-Rep.

Modèle volumique CSG (Constructive Solid Geometry)

Un modèle CSG, ou arbre CSG, est généralement construit à partir de primitives géométriques volumiques simples telles que des parallélépipèdes, des sphères ou des cylindres (Fig. 0.1.a). Le processus de construction est décrit par des opérations booléennes d'intersection, d'union ou de soustraction. Ce modèle est alors stocké sous forme d'arbre dont les feuilles sont des primitives géométriques et dont les noeuds représentent des opérations booléennes.

Ce type de représentation nécessite généralement peu de mémoire de stockage. Cependant, le modèle est construit à chaque ouverture de fichier en réalisant toutes les opérations booléennes nécessaires, ce qui peut être coûteux pour des objets complexes.

Modèle surfacique B-Rep (Boundary Representation)

Un modèle B-Rep est composé d'un ensemble de *faces*, définies par des primitives géométriques surfaciques telles que des plans, des sphères, des cylindres, ou des surfaces plus complexes (Fig. 0.1.b). Ces primitives surfaciques sont décrites sous une forme paramétrique, c'est-à-dire définies par un ensemble de paramètres ou des équations mathématiques. Un modèle B-Rep contient également des contours qui permettent de délimiter ces surfaces. Ces derniers sont alors décrits par des courbes paramétriques, et correspondent aux intersections entre les primitives adjacentes. Ce modèle est alors stocké sous forme d'un ensemble de primitives géométriques et d'un ensemble de contours. Notons qu'au delà des primitives géométriques, le modèle B-Rep peut aussi permettre de définir des surfaces plus complexes définies par une interpolation entre plusieurs courbes. Dans ce cas, nous parlons de surfaces “à forme libre”.

Par rapport au modèle CSG, ce type de représentation nécessite plus de mémoire pour stocker les contours construits. Mais dans ce cas, la construction du modèle à l'ouverture des fichiers est grandement simplifiée et prend alors beaucoup moins de temps.

Nous pouvons remarquer que ces deux représentations sont plutôt complémentaires. En effet, la plupart des logiciels de modélisation CAO présentent la construction d'un objet sous forme d'arbre, à l'image du modèle CSG, facilement interprétable par l'utilisateur. Mais lors de l'enregistrement des fichiers, il est préférable de stocker un modèle B-Rep. Les formats de fichier récents peuvent par ailleurs contenir des modèles B-Rep étendus, qui permettent d'enregistrer d'autres informations telles que l'adjacence entre les primitives par exemple.

Représentation discrète

La représentation discrète d'un objet manufacturé correspond en général à un maillage 3D. Ce dernier est construit à l'aide d'un ensemble fini de points, aussi désignés par le terme "sommets", qui sont reliés par des arêtes de manière à former des faces, aussi désignées par les termes "facettes" ou "éléments". La plupart du temps, nous construisons des faces triangulaires (trois points) ou quadrangulaires (quatre points). En règle générale, tout maillage peut être ramené à un maillage triangulaire, ce qui est d'ailleurs le cas lors de leur affichage par une carte graphique. Dans ce cas, nous nous concentrons sur les maillages triangulaires, qui sont les plus fréquemment rencontrés. Nous désignons alors une face par le terme "triangle". Cependant, ces maillages peuvent être construits de plusieurs manières, et posséder des caractéristiques différentes. Nous distinguons deux types de maillage : soit issu de discrétisation CAO, soit issu de numérisation 3D (Fig. 0.2).

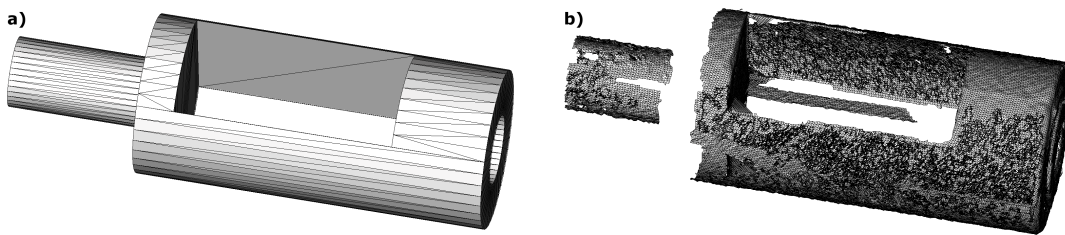


FIGURE 0.2 – Exemple de maillage issu de : a) Discrétisation CAO, b) Numérisation 3D.

Maillages issus de discrétisation CAO

Un maillage peut être construit par discrétisation d'une représentation continue. Dans ce cas, les points du maillage sont déterminés à partir d'une surface continue, et les coordonnées sont "exactes". Lorsque le modèle continu contient des primitives géométriques, les algorithmes de discrétisation s'appuient généralement sur leurs spécificités pour créer le moins de points possible. Par exemple, un plan carré peut être représenté par seulement quatre points formant deux triangles. Mais certaines primitives, comme la sphère, ne peuvent pas être discrétisées avec peu de points. Dans ce cas, des points sont répartis de manière relativement uniforme sur la surface. Cette seconde méthode de discrétisation est notamment utilisée lorsque le modèle paramétrique contient des surfaces dites "à forme libre", qui sont définies à partir de courbes 3D. Dans la plupart des cas, les maillages issus de discrétisation CAO contiennent généralement peu de points et sont dits de faible densité (Fig. 0.2.a).

Maillages issus de numérisation 3D

Un maillage peut aussi être obtenu à partir de la numérisation 3D d'un objet réel. Les scanners 3D peuvent utiliser différentes technologies telles qu'un balayage laser, une photogrammétrie ou encore la déformation d'un motif de lumière structurée [13]. Certaines de ces technologies fournissent directement le maillage grâce à des *a priori* sur le voisinage entre les points [12]. Pour les autres technologies, il est nécessaire d'appliquer un algorithme de reconstruction [11].

Ces dernières années, la numérisation 3D est devenue plus accessible. Cependant, la combinaison des différentes technologies et algorithmes ont engendré un grand nombre de caractéristiques différentes. Ainsi, les maillages que nous pouvons trouver peuvent être plus ou moins denses, plus ou moins précis, ou encore avoir subi des post-traitements. Dans la plupart des cas, ces maillages sont très denses et leurs coordonnées sont "inexactes" à cause des imprécisions de numérisation (Fig. 0.2.b).

En particulier, les points aux abords des arêtes vives d'un objet sont très imprécis. Des congés, qui sont des arêtes cylindriques qui adoucissent les bords d'un objet, se forment alors, même s'ils n'existent pas dans le modèle original (Fig. 0.3). En effet, la reconstruction du maillage se base sur les points obtenus, qui ont une très faible probabilité d'appartenir à une arête vive. Ces dernières ne peuvent donc pas être reconstruites de manière fiable.

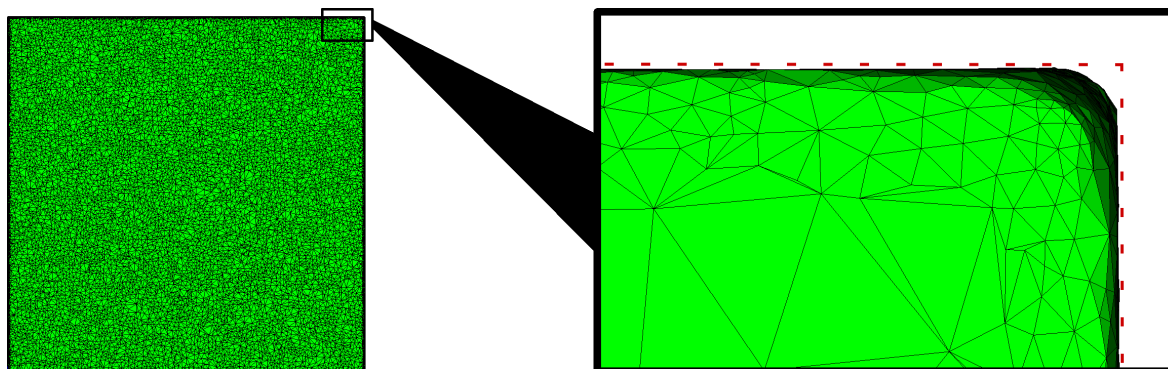


FIGURE 0.3 – Numérisation d'un cube : les imprécisions des points génèrent des congés (arêtes arrondies).

La rétro-ingénierie (*Reverse Engineering*)

Lors de leur conception numérique, les objets 3D sont généralement conçus à l'aide de logiciels de CAO. Ces logiciels permettent de construire des modèles dits "paramétriques", c'est-à-dire définis par une combinaison d'entités géométriques telles que des plans, des sphères, des cylindres ou des cônes. Ces modèles peuvent donc être représentés sous forme de CSG ou de B-Rep. D'un autre côté, la numérisation 3D fournit une représentation discrète d'un objet, sous forme de maillage 3D. Ce dernier ne contient pas directement d'information sur la géométrie de l'objet original.

Afin de retrouver le modèle paramétrique correspondant à un maillage, par exemple pour modifier l'objet ou analyser ses déformations, nous pouvons appliquer un processus de rétro-ingénierie. Ce dernier consiste à retrouver les paramètres des surfaces constituant l'objet, à partir de sa discrétisation 3D. Ensuite, le modèle initial peut être reconstruit. La figure 0.4 montre un exemple de conversion d'un maillage 3D, défini par un ensemble fini de points et de triangles, en un ensemble de primitives géométriques (plans en vert et cylindres en rouge) définies par des paramètres d'orientation, de position et de dimensions.

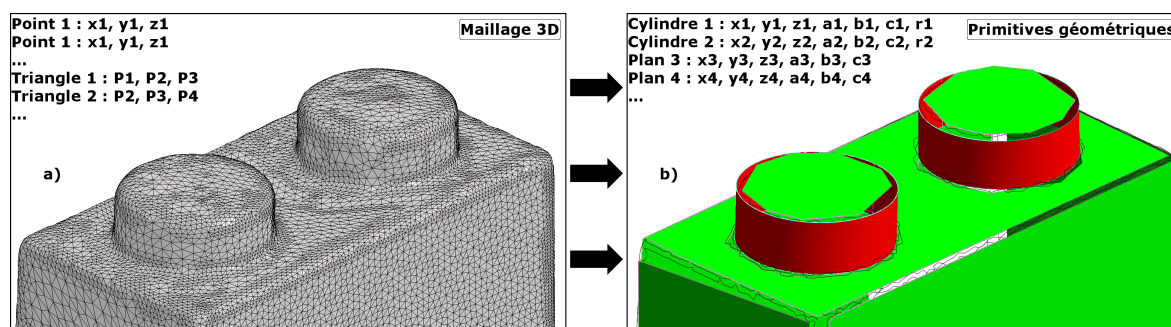


FIGURE 0.4 – Principe de la rétro-ingénierie : a) Maillage 3D issu de numérisation, défini par des points et des triangles, b) Primitives géométriques extraites, définies par des paramètres et des équations.

Positionnement de la thèse

L'entreprise C4W¹, basée à Montpellier, propose des logiciels de CAO traitant à la fois des objets discrets et continus. Dans le cadre d'une thèse CIFRE² en collaboration entre le LIRMM³ et l'entreprise C4W, nous avons étudié le problème de la rétro-ingénierie.

1. C4W : CAD for Windows (www.c4w.com)

2. CIFRE : Conventions Industrielles de Formation par la REcherche (www.anrt.asso.fr)

3. LIRMM : Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (www.lirmm.fr)

En particulier, une méthode de rétro-ingénierie à partir de maillages issus de discrétisation CAO a été développée lors de la thèse de Roseline Bénière [5]. L'entreprise a alors souhaité faire évoluer cette solution, en l'adaptant aux maillages issus de numérisation 3D.

Les travaux de thèse de Roseline Bénière [5] sont basés sur plusieurs *a priori* concernant les maillages issus de CAO. Par exemple, les coordonnées des points sont considérées exactes et les arêtes vives des objets sont bien marquées. Malheureusement, ces *a priori* ne sont pas valables pour les maillages issus de numérisation. En effet, ces derniers souffrent de nombreuses imprécisions dues aux procédés de fabrication et de numérisation, et peuvent même représenter des objets incomplets ou encore déformés avec le temps.

Cahier des charges

Pour répondre à une forte demande industrielle, nous nous intéressons principalement aux maillages triangulaires issus de numérisation 3D.

- Nous nous concentrons sur les maillages 3D d'objets manufacturés, qui contiennent principalement des primitives géométriques. Nous cherchons alors à extraire des plans, des sphères, des cylindres et des cônes.
- Les technologies ayant permis d'obtenir un maillage n'étant pas imposées, aucun *a priori* ne peut être fait en terme de résolution ou de précision.

Deux problèmes sont ainsi définis. Le premier consiste à extraire des primitives géométriques surfaciques de manière robuste à partir d'un maillage issu de numérisation. Le second problème consiste à corriger ou régulariser les primitives obtenues, dont les paramètres sont inexacts puisqu'elles proviennent au départ d'un maillage imprécis. L'idée est de pouvoir proposer au final un modèle paramétrique se rapprochant au mieux du modèle originalement conçu en CAO.

Puisqu'un maillage est une représentation discrète et surfacique d'un objet, l'objectif final est de pouvoir reconstruire un modèle continu surfacique. Dans ce cas, nous cherchons à reconstruire des modèles B-Rep à l'aide de la méthode proposée par Bénière *et al.* [5].

Plan des travaux développés

Ce manuscrit est composé de deux parties. Dans la première partie, nous présentons un état de l'art sur l'extraction des primitives à partir d'un maillage 3D (chapitre 1) et sur l'analyse du modèle reconstruit afin, par exemple, de le corriger (chapitre 2). Dans la seconde partie, nous décrivons nos contributions concernant ces deux problématiques (chapitres 3 et 4). Chaque chapitre montre également des résultats et se termine par une conclusion. Enfin, nous proposons une conclusion générale et des perspectives.

La figure 0.5 illustre le processus de rétro-ingénierie développé dans cette thèse. Celui-ci se décompose en deux parties :

- **Extraction de primitives (chapitre 3)**. Dans cette partie, nous cherchons tout d'abord à décomposer le maillage en sous-maillages correspondant à une unique primitive (Fig. 0.5.a). Ensuite, les sous-maillages obtenus sont analysés afin d'en déduire des paramètres de primitives de manière robuste (Fig. 0.5.b).
- **Analyse et correction des primitives (chapitre 4)**. Dans cette partie, nous cherchons à analyser les primitives obtenues d'un point de vue global. Cela permet d'en déduire des informations liées à la conception originale de l'objet, comme un système de référence non défini *a priori* (Fig. 0.5.c). Dans un second temps, ces informations sont utilisées pour corriger les paramètres des primitives extraites de sorte qu'elles respectent des relations géométriques comme le parallélisme, l'orthogonalité, la concentricité, la tangence, l'alignement selon une direction et l'égalité dimensionnelle (Fig. 0.5.d).

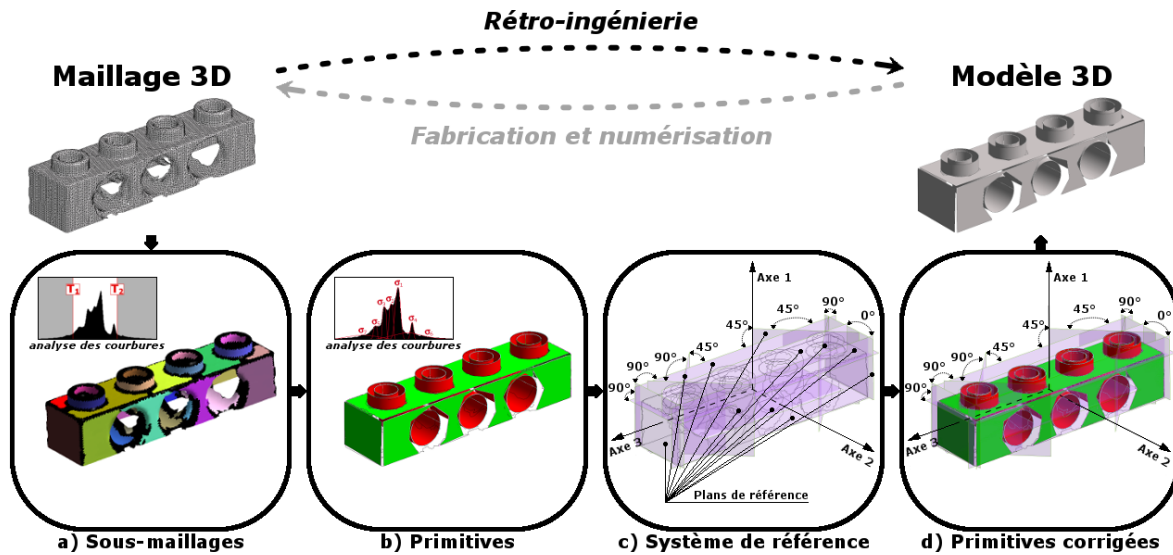


FIGURE 0.5 – Processus de rétro-ingénierie proposé : a) Décomposition du maillage 3D, b) Ajustement de primitives géométriques sur les sous-maillages obtenus, c) Analyse des primitives et extraction d'informations de conception, d) Correction des primitives afin qu'elles respectent des relations géométriques.

Première partie

Etat de l'art



Chapitre 1

Extraction de primitives géométriques à partir d'un maillage 3D

« Quand l'eau courbe un bâton, ma raison le redresse. »

Jean de La Fontaine

Résumé

L'extraction des primitives géométriques à partir d'un maillage 3D peut s'appuyer sur différentes approches. Ainsi, ces primitives peuvent être approximées directement à partir des points ou à partir d'indicateurs de forme locale tels que les normales ou les courbures.

Après une introduction sur l'extraction de primitives (section 1.1), nous présentons les deux approches principales proposées dans la littérature. Ainsi, l'approche incrémentale est détaillée dans la section 1.2. Ensuite, nous décrivons les trois principales étapes de celle-ci. La section 1.2.1 présente des méthodes de segmentation permettant de séparer les zones correspondant aux primitives géométriques. La section 1.2.2 décrit plusieurs méthodes caractérisant la forme du maillage à l'aide de la courbure. La section 1.2.3 présente des méthodes permettant d'ajuster des primitives sur des ensembles de points 3D. Enfin, l'approche directe est détaillée dans la section 1.3, suivie d'une synthèse (section 1.4).

Sommaire

1.1 Introduction	10
1.2 Méthodes incrémentales	12
1.2.1 Segmentation d'un maillage 3D en zones homogènes	16
1.2.2 Caractérisation locale de la forme 3D par courbure	27
1.2.3 Ajustement d'une primitive sur un nuage de points	32
1.3 Méthodes directes	35
1.4 Synthèse	38

1.1 Introduction

Nous avons vu dans le chapitre introductif que lors de leur conception, les objets manufacturés contiennent majoritairement des primitives géométriques simples. Un des principaux objectifs de la rétro-ingénierie est d'extraire ces primitives, ce qui est souvent réalisé selon une complexité croissante. Dans ces travaux, nous nous concentrons principalement sur quatre types de primitives que nous considérons comme les plus fréquemment utilisés, à savoir les plans, les sphères, les cylindres et les cônes (Fig 1.1).

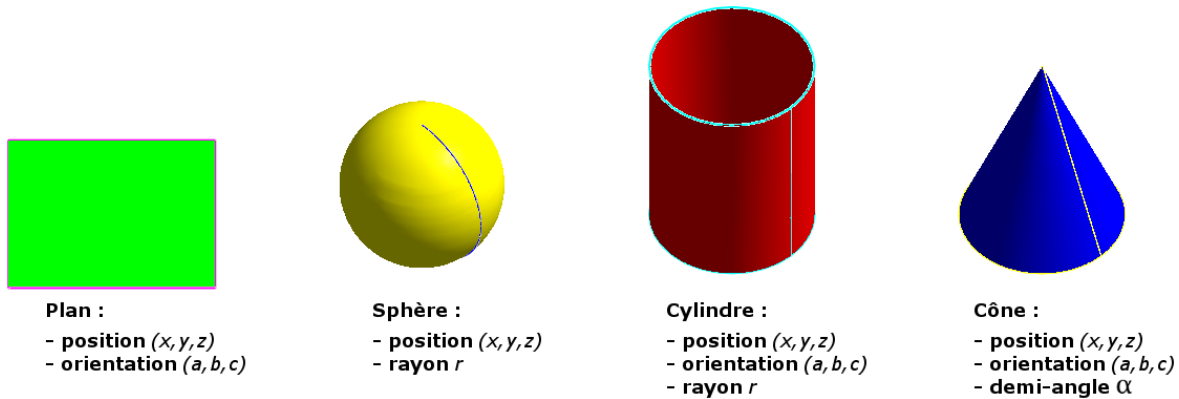


FIGURE 1.1 – Primitives géométriques considérées comme fréquemment utilisées, avec leurs paramètres.

Petitjean [86] a proposé un état de l'art des méthodes d'extraction de primitives géométriques, à partir d'un maillage 3D. Cet état de l'art détaille les différentes étapes du processus d'extraction. La première concerne la définition de la forme locale à travers des descripteurs comme les normales ou les courbures. Ensuite, l'auteur décrit des approches permettant de déterminer le type de primitive associé aux points. Puis, il propose des algorithmes de pré-traitement pour améliorer les résultats. La quatrième partie est un état de l'art des méthodes de segmentation de maillages. Cette partie, relativement longue, suggère que l'étape de segmentation est importante. Enfin, des méthodes permettant d'ajuster une primitive sur un ensemble de points sont présentées, en particulier des régressions linéaires.

Bradley et Currie [14] présentent un autre état de l'art sur les méthodes d'extraction de primitives, en détaillant les différentes étapes nécessaires. En premier lieu, une segmentation du maillage est effectuée afin de séparer les zones correspondant aux primitives géométriques. Une fois le maillage segmenté, un ajustement est réalisé afin d'estimer les paramètres des primitives présentes dans l'objet, en particulier leur orientation, leur position et leurs dimensions. Des approches guidées par l'utilisateur à chaque étape sont également décrites, mais celles-ci sont plus contraignantes du fait qu'elles impliquent une certaine expertise de ce dernier. Enfin, les auteurs détaillent plusieurs exemples d'applications tels que l'art, l'imagerie médicale et dentaire, ou encore le développement (ou re-développement) de produits manufacturés. De ce fait, ils pointent le besoin de concevoir des méthodes les plus génériques possibles.

Plus récemment, Buonamici *et al.* [18, 17] proposent deux états de l'art des méthodes d'extraction de primitives. Ils présentent ce processus comme une reconstruction en plusieurs étapes (Fig. 1.2). La première concerne d'éventuels prétraitements tels que le lissage, la décimation ou encore le remaillage. Ensuite, le maillage est segmenté en plusieurs régions, puis chaque région est caractérisée par un type de primitive géométrique. Ils expliquent en particulier que cette classification est souvent liée à la segmentation, notamment parce que ces deux étapes définissent souvent la géométrie sous-jacente de la même manière. Enfin, les primitives sont ajustées sur les régions correspondantes. Concernant cet ajustement, les auteurs décrivent quatre types d'approche : indépendante, contrainte, à partir de sections 2D, et à partir des connaissances de l'utilisateur. Concernant la dernière approche, qui nécessite régulièrement l'action d'un utilisateur expert, les auteurs pointent le besoin de concevoir des méthodes automatiques.

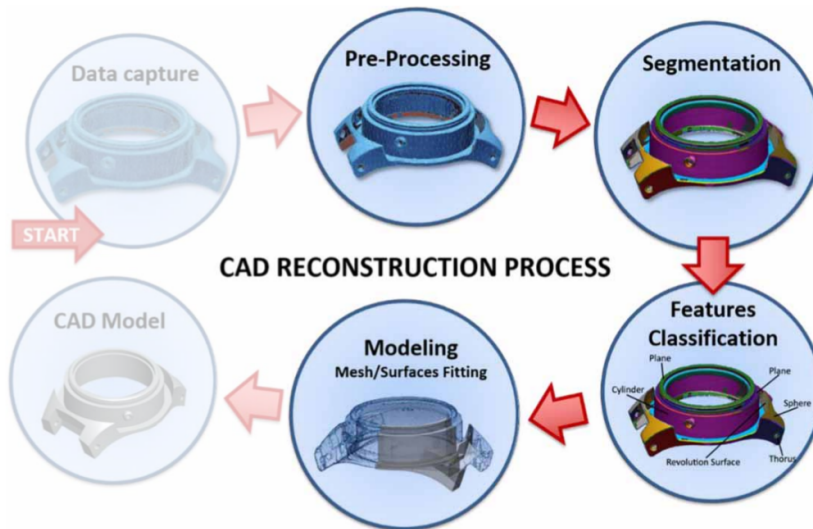


FIGURE 1.2 – Processus d'extraction de primitives en quatre étapes présenté par Buonamici *et al.* [17].

Nous pouvons remarquer que la plupart des méthodes d'extraction de primitives utilisent en premier lieu une segmentation du maillage. En effet, les descripteurs utilisés par la suite pour caractériser la forme locale sont souvent calculés à partir d'un voisinage sur les points. Par exemple, le calcul de la courbure, un indicateur de forme détaillé dans la section 1.2.2, est basé sur le k -voisinage de chaque point P . Ce dernier correspond à l'ensemble des points du maillage reliés à P par une séquence d'au plus k arêtes. Or, ce voisinage est déterminant lorsqu'un maillage contient des données imprécises, ce qui est généralement le cas pour ceux issus de numérisation. Dans la figure 1.3, on remarque qu'un voisinage de 1 ne permet pas d'estimer des courbures planes sur un maillage imprécis. De même, un voisinage de 10 est trop élevé, ce qui tend à réduire ou éliminer les zones planes proches des arêtes de l'objet. En effet, il est à noter que plus le voisinage est grand, plus les imprécisions sont atténuées. Cependant, ceci implique également d'augmenter le risque que ce voisinage contienne des points issus de plusieurs primitives différentes, faussant alors les valeurs obtenues. L'étape de segmentation permet ainsi d'éliminer ce risque, en limitant le voisinage à une certaine région.

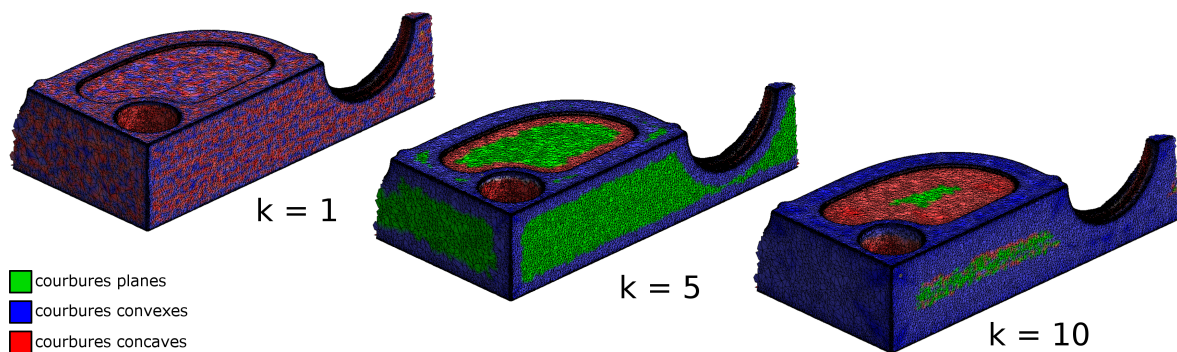


FIGURE 1.3 – Impact du k -voisinage des points pour le calcul de la courbure.

Synthèse

Dans la littérature, nous distinguons deux types d'approches pour l'extraction de primitives. Le premier type d'approche, correspondant à la plupart des méthodes, se déroule en plusieurs étapes : une segmentation du maillage, une caractérisation de la forme et un ajustement des primitives. Dans la suite, nous utilisons le terme de "méthodes incrémentales". Concernant le second type d'approche, proposant d'extraire directement les primitives à partir du maillage en une seule étape d'ajustement, nous utilisons le terme de "méthodes directes".

1.2 Méthodes incrémentales

Les méthodes incrémentales procèdent en plusieurs étapes. Cela permet de découper le problème d'extraction de primitives géométriques en plusieurs sous-problèmes, à traiter les uns après les autres. Ainsi, extraire une primitive consiste à réaliser les trois étapes suivantes.

1. Définir les régions correspondant aux différentes primitives.
2. Classifier les régions en caractérisant leur forme.
3. Approximer une primitive à partir de chaque région.

Une des premières méthodes incrémentales a été proposée par Benkő *et al.* [10]. L'algorithme commence par estimer une normale pour chaque point en ajustant un plan sur les points voisins. Ensuite, le maillage est segmenté au niveau des changements brutaux de direction des normales, selon un seuil. Cela permet d'obtenir des sous-maillages distincts, qui sont alors catégorisés soit en régions *arêtes* (changements brutaux), soit en régions *primitives* (zones homogènes). Un ajustement est alors effectué sur chaque sous-maillage *primitive* pour rechercher une primitive simple. Si cet ajustement n'est pas concluant, un plan est choisi et le profil du sous-maillage sur ce plan est extrait (Fig. 1.4), à partir de segments et arcs de cercle. Une surface complexe (extrusion ou révolution) est ensuite reconstruite à partir du profil obtenu. Nous pouvons noter l'importance de la séparation entre les zones, correspondant d'un côté aux arêtes et de l'autre aux primitives. En effet, nous avons vu dans l'introduction de ce manuscrit que dans un maillage issu de numérisation, les points sont souvent très imprécis au niveau des arêtes vives de l'objet.

Cependant, la segmentation utilisée ne permet pas de séparer correctement toutes les primitives, ce qui pose par la suite un problème d'identification. De plus, l'approche basée sur les profils semble limitée aux extrusions ou aux surfaces de révolution.

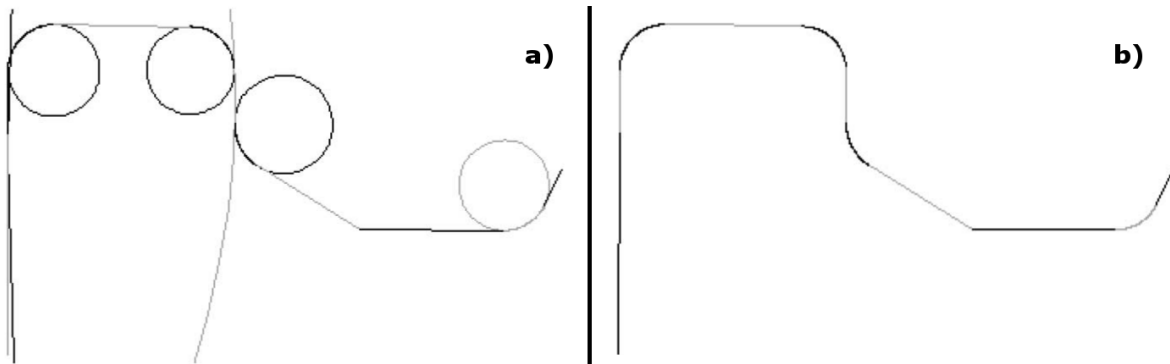


FIGURE 1.4 – Résultats de Benkő *et al.* [10] : a) Segments et arcs de cercle ajustés, b) Profil reconstruit.

Huang et Menq [48] proposent une méthode en trois étapes (Fig. 1.5). D'abord, un maillage triangulaire est construit à partir d'un nuage de points. Ensuite, une segmentation est appliquée, puis une primitive est approximée sur chaque sous-maillage. Si un sous-maillage ne correspond à aucune primitive simple, alors une surface paramétrique à base de NURBS est calculée.

Cependant, nous montrons en section 1.2.1 que l'approche de segmentation utilisée n'est pas très adaptée à l'extraction de primitives et aux objets complexes. Or, si la segmentation n'est pas suffisamment performante, les primitives ne sont pas correctement séparées. Par conséquent, des sous-ensembles de primitives voisines peuvent être approximées par une seule et même surface, ce qui pose alors un problème d'identification.

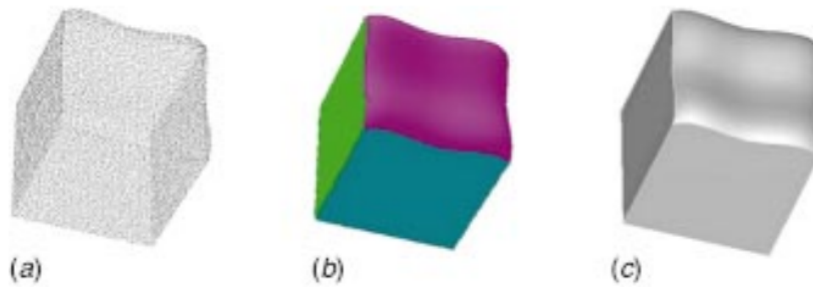


FIGURE 1.5 – Résultats de Huang et Menq [48] : a) Points 3D, b) Sous-maillages, c) Primitives et NURBS.

La méthode de Wang *et al.* [113] se décompose en trois étapes (Fig. 1.6). Premièrement, une segmentation est réalisée après un débruitage préalable du maillage permettant de réduire l'imprécision des données tout en conservant au mieux la forme. Cette segmentation procède hiérarchiquement en considérant chaque zone selon un type de primitive, dans un ordre croissant de complexité (plan, cylindre, cône, sphère...). Si la zone ne correspond à aucune des primitives données, alors elle est segmentée à nouveau. Deuxièmement, des primitives sont approximées à partir des sous-maillages obtenus. Troisièmement, les primitives sont ajustées en imposant des contraintes de continuité entre elles, afin d'obtenir un modèle fermé. Ce processus complet met en exergue les étapes importantes de l'extraction de primitives. Notons en particulier que la récursivité permet de segmenter le maillage et d'extraire des primitives conjointement, de manière adaptative.

Cependant, les résultats obtenus sont relativement dépendants de l'étape de pré-traitement, qui nécessite d'être correctement paramétrée. De même, les étapes suivantes ne sont pas entièrement automatiques et nécessitent donc un utilisateur expert tout au long du processus.

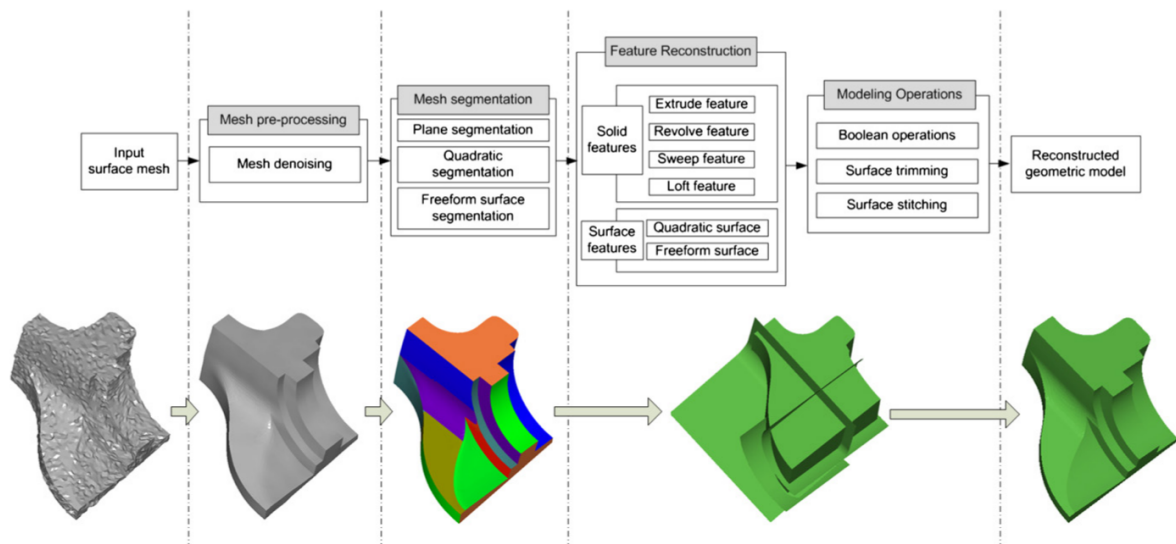
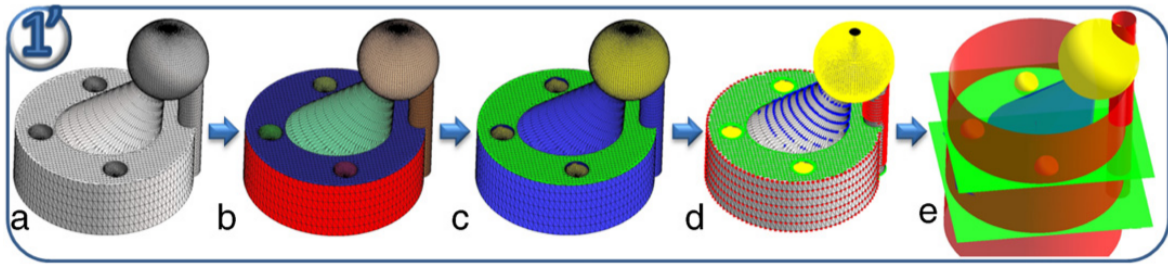


FIGURE 1.6 – Schéma global de la méthode de Wang *et al.* [113].

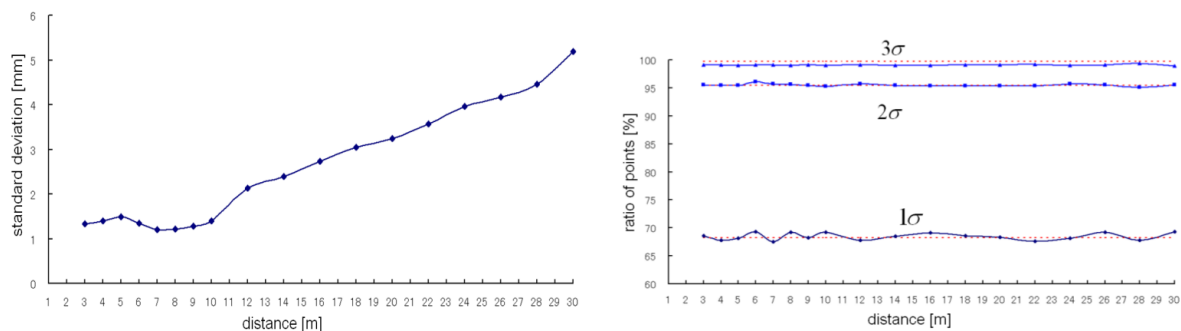
La méthode de Bénére *et al.* [8] commence par segmenter le maillage (Fig. 1.7.b), puis calcule la courbure sur chaque sous-maillage (Fig. 1.7.c). Des zones de points sont alors définies localement en fonction des valeurs de courbure, en déterminant le type de primitive pour chaque point (Fig. 1.7.d). Ensuite, une primitive est ajustée sur chaque zone de points en fonction de son type (Fig. 1.7.e).

Cependant, cette méthode est basée sur de nombreux *a priori* concernant les maillages issus de discrétisation CAO, ce qui ne permet pas de l'utiliser sur des maillages issus de numérisation.


 FIGURE 1.7 – Schéma global de la méthode de Bénéière *et al.* [8].

Masuda *et al.* [70] proposent une méthode d'extraction directement incluse dans le processus de numérisation. Ainsi, le type et les caractéristiques du scanner utilisé permettent d'améliorer l'extraction de primitives en définissant des poids sur les points en fonction de leur degré d'imprécision. En effet, pour les scanner laser utilisés, l'imprécision contenue dans les points 3D est liée à plusieurs paramètres, tels que la distance du point par rapport au scanner, la réflectance de l'objet, ou encore son inclinaison. En mesurant les déviations sur une mire, il est possible de définir un modèle de répartition des déviations (Fig. 1.8). Ce modèle peut alors être utilisé pour déterminer la fiabilité de chaque point et ainsi adapter la segmentation et le calcul des primitives.

Cependant, cette méthode dépend intrinsèquement du scanner utilisé, et n'est présentée que pour des scanners laser. Nous pouvons alors nous demander si cette méthode est adaptable pour d'autres types de scanner. De plus, les auteurs ne présentent que l'extraction de plans et cylindres, ce qui limite encore les cas d'utilisation.


 FIGURE 1.8 – Mesures des imprécisions réalisées par Masuda *et al.* [70].

Chen et Feng [22] proposent d'utiliser une extraction de primitives afin d'*idéaler* des maillages issus de numérisation (Fig. 1.9.a). Cette *idéation* consiste à projeter les points du maillage sur les primitives extraites, et ainsi obtenir des coordonnées exactes. Tout d'abord, une segmentation de l'objet est appliquée en se basant sur la courbure. Ensuite, des primitives géométriques sont extraites en fonction d'un ordre prédéfini (plans, cylindres, cônes, sphères, et enfin tores), censé refléter l'ordre de conception CAO initial de la pièce. Les primitives extraites sont ensuite ajustées afin de respecter des contraintes géométriques en fonction du plus grand plan détecté (cette partie est détaillée dans le chapitre 2). Enfin, chaque point du maillage est projeté sur la primitive correspondante dans le but d'obtenir des coordonnées *exactes*. Les arêtes vives sont également reconstruites en alignant les points sur les intersections entre primitives voisines.

Cependant, comme détaillé en section 1.2.1, la segmentation utilisée ne permet pas toujours de bien séparer les primitives. D'un autre côté, l'ordre de priorité défini sur les primitives ne peut pas correspondre à toutes les possibilités de conception d'une pièce en CAO. Enfin, l'ajustement des primitives en se basant sur un plan de référence imprécis ne permet pas de s'assurer de la qualité globale du modèle final.

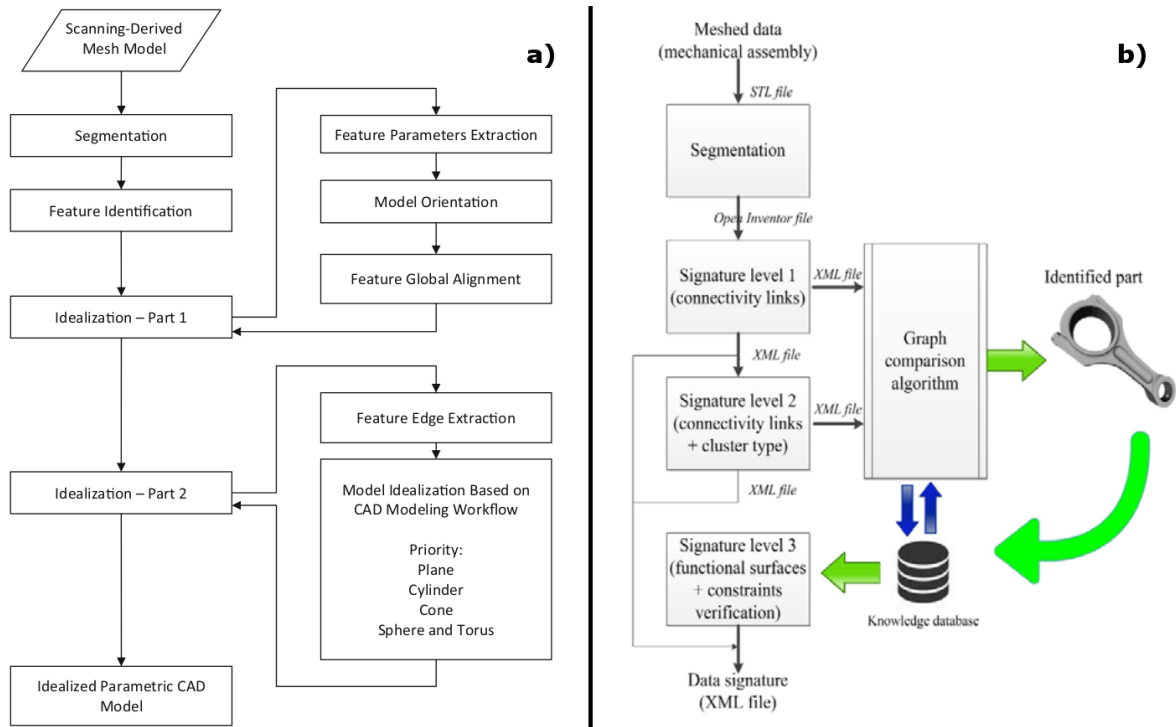


FIGURE 1.9 – Schémas globaux des méthodes de : a) Chen et Feng [22], b) Bruneau *et al.* [15].

Dans un autre contexte, Bruneau *et al.* [15] proposent un algorithme permettant de reconnaître des assemblages à partir de signatures, basées sur la construction de graphes successifs (Fig. 1.9.b). Dans un premier temps, le maillage est segmenté. Un premier graphe est alors construit à partir du voisinage entre les sous-maillages obtenus. Ensuite, le type de primitive est déterminé pour chaque sous-maillage, ce qui permet de construire un second graphe. Puis, un troisième graphe est construit à partir des contraintes géométriques entre les primitives, censées refléter les conditions d'assemblage des pièces manufacturées. Au final, une base de données est construite avec de nombreux graphes calculés à partir de différents modules d'assemblage typiques. Enfin, cette base de données peut être interrogée pour déterminer quels modules constituent une pièce donnée et avec quelles contraintes.

Malheureusement, cette approche demande de nombreuses comparaisons de graphes, qui peuvent parfois être très complexes. De plus, la base de données utilisée est construite à partir de connaissances *a priori* sur les modules utilisés dans les assemblages. Enfin, cette base de données doit être exhaustive pour être efficace sur tous les objets manufacturés.

Synthèse

Les méthodes incrémentales assurent en général une bonne couverture du maillage par des primitives. Cependant, elles sont souvent assez sensibles aux imprécisions, en particulier pour les approches calculant des descripteurs de formes avec un voisinage trop restreint, ne permettant alors pas d'atténuer le bruit. Il est également important de noter que l'étape de segmentation semble déterminante pour obtenir de bons résultats. De même, l'utilisation conjointe de la segmentation et de l'ajustement de primitives de manière récursive semble permettre une meilleure robustesse lorsque les primitives sont de différentes tailles ou lorsque les imprécisions ne sont pas uniformément distribuées dans le maillage. Enfin, il semble que la courbure soit l'indicateur de forme le plus adapté à l'extraction de primitives.

1.2.1 Segmentation d'un maillage 3D en zones homogènes

Une segmentation est un partitionnement d'une image, d'un maillage ou d'un nuage de points en plusieurs régions ou sous-ensembles distincts. Il existe de nombreuses approches différentes, qui sont souvent adaptées à des applications particulières. Notre problématique étant d'utiliser une segmentation pour améliorer l'extraction de primitives, l'approche consistant à segmenter un maillage en se basant sur les primitives extraites ne sera pas détaillée.

Chen *et al.* [23] proposent une méthodologie permettant de comparer plusieurs algorithmes de segmentation de maillages 3D, en les appliquant à une même base de données. Cette base de données contient des maillages regroupés en 19 catégories : des pièces manufacturées, des sculptures, des peluches, des animaux, des véhicules, des êtres humains... etc. Différentes approches de segmentation sont alors appliquées à toutes ces catégories. Cela pose un réel problème quant à la comparaison des méthodes de segmentation, qui ne permettent souvent de segmenter correctement qu'une seule catégorie d'objets. De plus, vis-à-vis de notre problématique, la base contient malheureusement peu d'objets manufacturés numérisés.

Shamir [97] propose un état de l'art des méthodes de segmentation de maillages 3D. Pour commencer, il existe selon lui trois types de contraintes pouvant être imposées sur les sous-maillages issus de la segmentation :

- la cardinalité, définie par le nombre de points ou de triangles;
- la géométrie, définie par un descripteur de forme;
- la topologie, définie par la connectivité des points et des triangles.

Celles-ci permettent d'homogénéiser les sous-maillages obtenus, par exemple en supprimant ceux qui sont trop petits. Ensuite, il existe une multitude de critères permettant de regrouper les éléments du maillage entre eux, par exemple une géométrie ou des dimensions similaires. Enfin, les méthodes de segmentation sont regroupées en deux catégories :

- les segmentations par patchs, cherchant à répartir au mieux l'aire de chaque région afin de couvrir le maillage de manière relativement homogène (Fig. 1.10.a);
- les segmentations par parties, capables de segmenter un maillage selon une logique de construction, par exemple en recherchant des membres anatomiques (Fig. 1.10.b).

Theologou *et al.* [104] propose également une troisième catégorie, qui regroupe les approches cherchant à définir une logique de segmentation commune entre plusieurs objets (Fig. 1.10.c). Dans les trois catégories, nous pouvons trouver plusieurs approches algorithmiques : par régions, par frontières, hiérarchiques, supervisées, interactives ou automatiques. Les auteurs décrivent ensuite plus précisément quelques méthodes de segmentation basées sur la définition de régions ou de frontières, l'ajustement de surfaces, la topologie, la division de l'espace, l'analyse spectrale, un modèle probabiliste ou encore la co-segmentation (Fig. 1.10.c). Ces méthodes sont alors évaluées avec la méthodologie de Chen *et al.* [23], dont la base de données n'est pas très adaptée à notre problématique, comme expliqué précédemment.

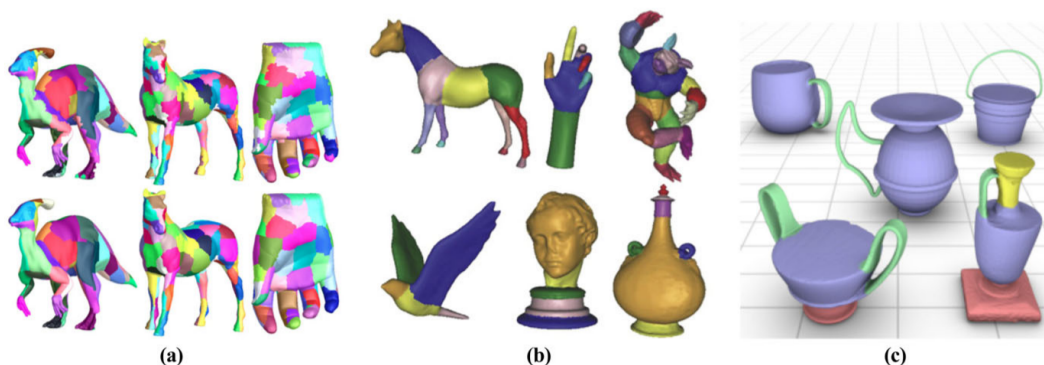


FIGURE 1.10 – Segmentations [89] : a) Par patchs [90], b) Par parties [111], c) Co-segmentation [98].

Rodrigues *et al.* [89] proposent un autre état de l'art des méthodes de segmentation, mais cette fois-ci axé autour des méthodes par parties (Fig. 1.10.b). Ces segmentations sont regroupées en trois catégories (Fig. 1.11).

Les premières, basées sur une décomposition volumique, ont pour objectif de diviser un objet en un ensemble de sous-volumes plus simples. Celles-ci semblent particulièrement adaptées au traitement des assemblages. Cependant, il semble que ces méthodes impliquent souvent de lourds calculs, ce qui peut être problématique pour les maillages très denses.

La seconde catégorie de méthodes contient celles cherchant à regrouper les éléments du maillage selon des critères géométriques. Celles-ci permettent généralement de retrouver des parties qui ont un sens du point de vue de l'objet, telles que les membres d'un être vivant ou les régions homogènes d'une pièce manufacturée. Cependant, ces méthodes demandent souvent de connaître le nombre de régions, ce qui pose problème pour les objets complexes.

Enfin, la dernière approche est basée sur la construction d'un squelette de l'objet, constitué d'un ensemble de segments connectés et représentant la structure de l'objet. Ces méthodes sont particulièrement adaptées aux structures tubulaires comme les membres d'un être humain par exemple, mais permettent rarement de segmenter correctement des objets mécaniques ou manufacturés. En effet, ces derniers n'ont généralement pas de squelette à proprement parler. De plus, ces méthodes ne détectent pas les éléments structurels secondaires n'étant pas suffisamment représentatifs. Or, notre objectif est de séparer toutes les régions correspondant aux primitives.

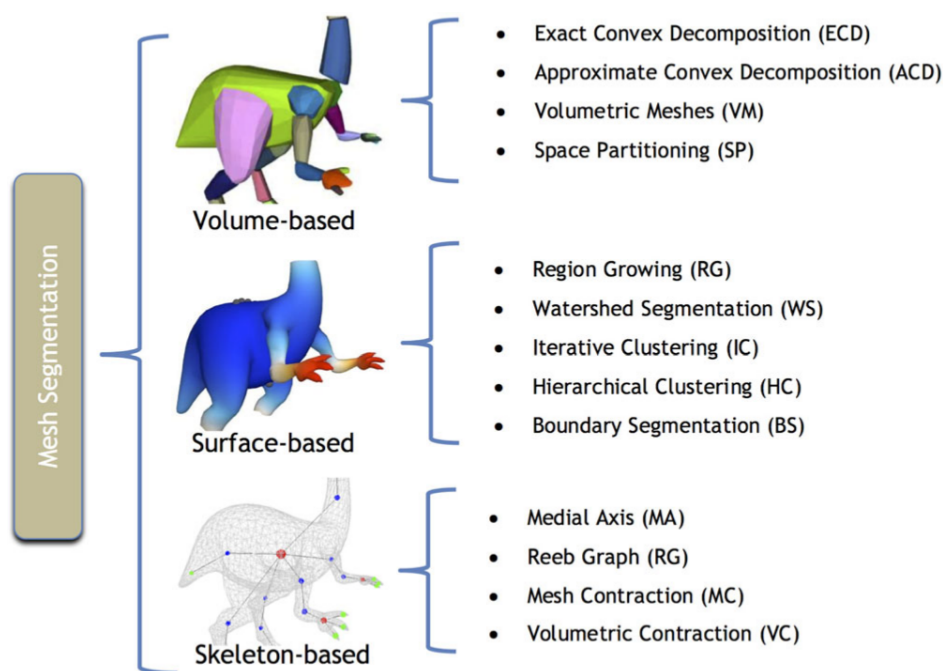


FIGURE 1.11 – Segmentations par parties regroupées en trois catégories [89].

Rashad *et al.* [88] présentent une hiérarchie détaillée de critères de segmentation, ainsi que plusieurs approches de segmentation par parties (Fig. 1.11, seconde catégorie). La première, procédant par croissance de régions, peut regrouper les éléments par propagation à partir d'un ou plusieurs éléments sources. Une variante, appelée *montée des eaux*, est basée sur la détection des *creux* d'un objet [26]. La métaphore suggère que si l'on remplit progressivement un *creux* avec de l'eau, l'endroit où l'eau se met à déborder indique sa limite et donc la frontière de la région. La seconde, procédant par fusion de régions, permet de mieux maîtriser l'ordre de regroupement des éléments. La troisième est une technique itérative permettant d'optimiser la répartition des éléments du maillage au sein d'un nombre prédéterminé de régions. Le problème de cette dernière est qu'elle nécessite de connaître le nombre de régions, ce qui semble difficile à estimer pour des pièces complexes.

Synthèse

De nombreux algorithmes de segmentation ont été proposés ces dernières années, mais chaque approche de segmentation dépend de l'application visée. Notre objectif étant de séparer les zones correspondant aux primitives de l'objet, les méthodes les plus adaptées semblent être celles basées sur les parties. Dans la suite de ce chapitre, ces méthodes sont présentées en deux catégories (Fig.1.12).

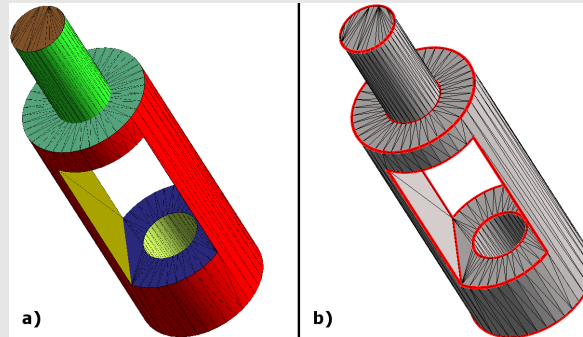


FIGURE 1.12 – Segmentation par définition de : a) Régions, b) Frontières.

Ainsi, nous présentons dans un premier temps les méthodes de segmentation cherchant à définir des régions homogènes. Ensuite, nous présentons celles cherchant à définir les frontières entre les régions, souvent définies par des lignes caractéristiques.

1.2.1.1 Définition de régions

Comme nous avons pu voir à travers plusieurs états de l'art sur la segmentation, nous pouvons distinguer trois types de construction des régions : par itérations, par croissance et par fusion.

Par itérations

Une première approche consiste à définir au préalable un nombre de régions, puis d'itérer un algorithme permettant de répartir au mieux les éléments du maillage au sein de ces régions (Algorithme 1). Ainsi, l'algorithme se termine lorsqu'il ne peut plus améliorer le partitionnement. Cette approche est souvent très stable, mais nécessite de définir le nombre optimal de régions au préalable, ce qui semble difficile à déterminer pour les objets complexes.

Algorithme 1 Partitionnement itératif sur les triangles

```

Créer un ensemble de régions R
Créer K régions et les ajouter à R
répéter
  pour tout triangle  $T_i$  du maillage faire
    Trouver la meilleure région candidate  $R_c$  pour  $T_i$ 
    Assigner  $T_i$  à  $R_c$ 
  fin pour
  pour tout région  $R_i$  de R faire
    Mettre à jour les caractéristiques de  $R_i$ 
  fin pour
jusqu'à ce que les régions soient stables
renvoyer R

```

Un des algorithmes de référence pour cette approche est celui des *K-Means* [64]. Il commence avec K barycentres, qui peuvent être choisis aléatoirement. Chaque point 3D est alors assigné à la partition dont le barycentre est le plus proche, en fonction d'une distance. Puis, chaque barycentre est recalculé en fonction des points assignés à la partition correspondante. Le processus est réitéré jusqu'à ce que tous les points soient assignés aux mêmes partitions qu'à l'itération précédente.

Lai *et al.* [55] présentent une méthode de segmentation utilisant les *K-Means* (Fig. 1.13). Dans un premier temps, le maillage initial est remaillé par l'utilisateur, afin de jouer sur la densité de celui-ci (Fig. 1.13.b). Ensuite, les *K-Means* sont appliquées sur l'objet remaillé avec une distance euclidienne. Enfin, les régions sont reconstruites sur le maillage initial en fonction de celles déterminées sur le maillage remaillé (Fig. 1.13.c). L'étape de remaillage permet ainsi de faciliter la recherche de régions homogènes en adaptant la densité du maillage.

Cependant, de nombreux paramètres sont demandés à l'utilisateur, dont des tolérances de distance pour le remaillage et le nombre de régions pour les *K-Means*. Cela implique une certaine expertise vis-à-vis du remaillage et de la segmentation.

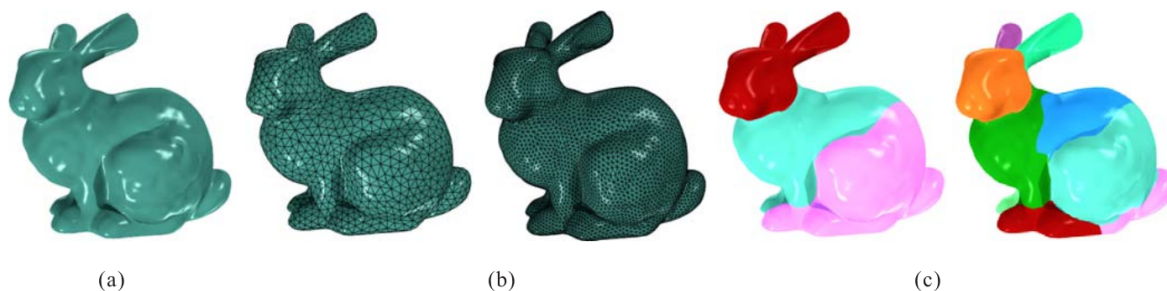


FIGURE 1.13 – Résultats de Lai *et al.* [55] : a) Maillage, b) Rémaillages, c) Segmentations.

Yi *et al.* [121] proposent d'améliorer les travaux de Gelfand *et al.* [42], qui ont proposé un descripteur de formes basé sur des valeurs propres de matrices de rotations et translations. Ils proposent d'utiliser une croissance de régions, dans un système itératif, afin de reconnaître et améliorer les régions correspondant à des primitives géométriques. Les contours sont par la suite rectifiés afin d'obtenir de meilleures frontières.

Cependant, lorsque toutes les valeurs propres associées à un point sont très élevées, le descripteur n'est pas défini et le point n'est pas catégorisé. De plus, l'estimation des normales en chaque point et le voisinage utilisé pour les calculs ont un fort impact sur les résultats (Fig. 1.14). Notons tout de même que les arêtes vives des objets semblent correctes dans tous les cas. Enfin, la méthode itérative implique de connaître le nombre de régions au préalable.

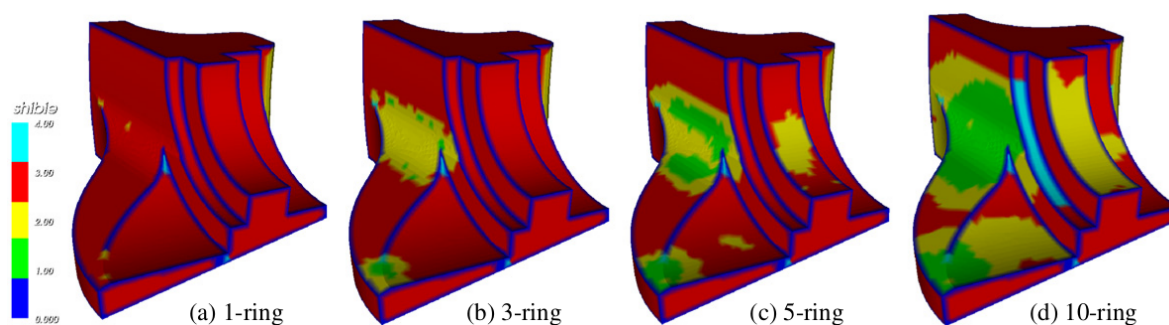


FIGURE 1.14 – Impact du voisinage sur le descripteur utilisé dans la méthode de Yi *et al.* [121].

Synthèse

Les méthodes itératives sont certes robustes aux imprécisions, mais demandent en entrée le nombre de partitions à réaliser, et parfois d'autres paramètres. Or, nous ne pouvons pas toujours connaître à l'avance le nombre de primitives contenues dans un objet, notamment pour les objets complexes. Il semble donc difficile de définir un nombre de régions pour la segmentation. De plus, ces méthodes impliquent souvent des temps de calcul importants.

Par croissance

Les algorithmes basés sur une croissance de régions définissent un élément de départ appelé *graine*, à partir duquel les éléments voisins vont être agglomérés (Algorithme 2). Les différents éléments sont ainsi regroupés en fonction de caractéristiques communes. Enfin, lorsqu'une région ne peut plus croître, une nouvelle graine est choisie, et le processus est réitéré jusqu'à avoir catégorisé tous les éléments du maillage. La difficulté de cette méthode concerne le choix de la graine, qui peut parfois grandement influencer le résultat en fonction des tolérances définissant si deux éléments peuvent être regroupés ou non. Certaines approches proposent de choisir plusieurs graines afin de réduire l'impact de ce choix.

Algorithme 2 Croissance de régions sur les triangles

```

Créer un ensemble de régions R
tant que il existe un triangle T non traité faire
  Créer une nouvelle région  $R_i$  et y ajouter T
  pour tout triangle  $T_i$  de  $R_i$  faire
    Marquer  $T_i$  comme traité
    pour tout triangle  $T_v$  non-traité et voisin de  $T_i$  faire
      si ( $T_i, T_v$ ) peuvent être regroupés alors
        Ajouter  $T_v$  à  $R_i$ 
      fin si
    fin pour
  fin pour
  Ajouter  $R_i$  à R
fin tant que
renvoyer R

```

Lavoué *et al.* [58] proposent un regroupement en fonction des valeurs de courbure du maillage (Fig. 1.15.a). Dans un premier temps, les arêtes vives sont détectées afin de limiter le voisinage utilisé dans le calcul de la courbure (voir section 1.1). Ensuite, les valeurs de courbure calculées sont catégorisées avec l'algorithme des *K-Means* et chaque point est associé à un des K groupes. Enfin, une croissance de régions est appliquée afin de regrouper tous les points voisins appartenant au même groupe. Nous pouvons remarquer que la catégorisation des points permet de ne pas avoir le problème du choix de la graine. Les auteurs proposent également un processus de raffinement des contours afin de rectifier les résultats (Fig. 1.15.b).

Cependant, cette méthode est conçue pour des maillages issus de discrétisation CAO et nécessite un utilisateur expert à même de définir le nombre K de groupes à utiliser.

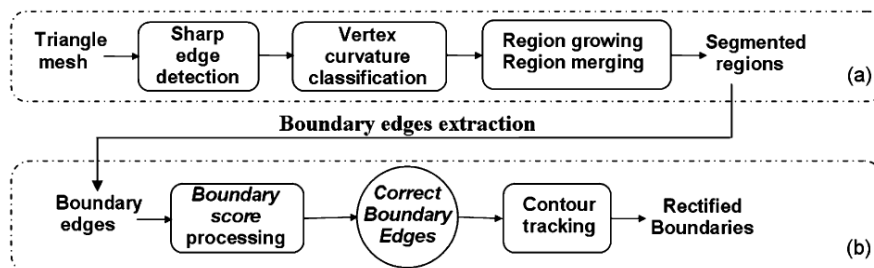


FIGURE 1.15 – Schéma global de la méthode de Lavoué *et al.* [58].

Mizoguchi *et al.* [74] proposent également d'utiliser la courbure. Tout d'abord, une reconnaissance des arêtes est réalisée par seuillage de la courbure selon un paramètre défini par l'utilisateur. Ensuite, chaque point est catégorisé en fonction d'un type de primitive, déterminé à partir des valeurs de courbure. Des primitives sont alors ajustées sur les régions obtenues. Enfin, des extrusions sont reconnues à partir des plans et cylindres extraits.

Cependant, les résultats obtenus sont directement liés aux choix faits par l'utilisateur. Cette méthode n'est donc pas automatique et dépend d'un utilisateur ayant besoin de connaissances approfondies sur la courbure et les primitives géométriques. De plus, les extrusions extraites ne correspondent pas toujours à la conception originale. En effet, une extrusion complexe n'est pas toujours réalisée en une seule fois mais peut être, par exemple, le résultat de plusieurs extrusions simples en intersection (Fig. 1.16.c), ce qui n'apparaît pas dans cette méthode.

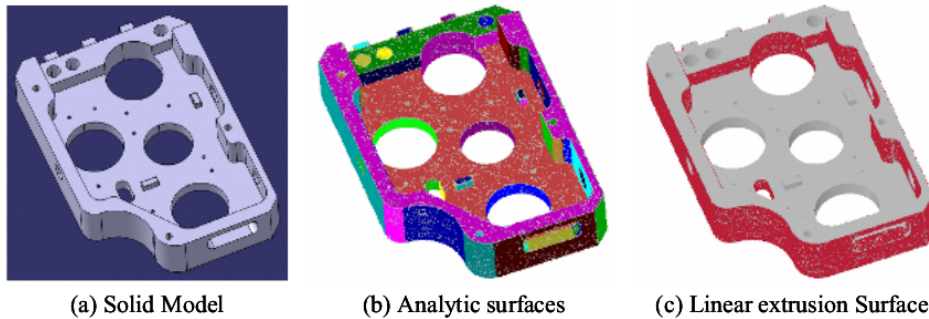


FIGURE 1.16 – Résultats de Mizoguchi *et al.* [74].

Bénière *et al.* [5] utilisent les angles dièdres entre les triangles voisins afin de construire des régions. Ainsi, un triangle *graine* est choisi et les angles dièdres formés avec ses voisins sont calculés. Chaque voisin est alors aggloméré à la région s'il forme un angle proche de 0° , selon une tolérance prédéfinie. Cela permet de séparer les régions au niveau des arêtes vives de l'objet.

Cependant, cette méthode n'est pas adaptée aux maillages issus de numérisation. En effet, les angles dièdres sont très variables à cause des imprécisions locales. De plus, cette segmentation ne permet pas de séparer les zones correspondant à des primitives tangentes par exemple, qui sont séparées par des arêtes lisses.

Di Angelo et Di Stephano [30] proposent de segmenter un maillage issu de numérisation grâce à un indicateur de forme proche de la courbure. Ce dernier permet de discriminer les points selon s'ils sont *réguliers* ou *non-réguliers*. Les points *non-réguliers* correspondent aux arêtes vives de l'objet et sont considérés à part pour la construction des régions. Pour les autres points, l'analyse des valeurs de courbure est basée sur le *shape index*, qui définit à quel point la surface est convexe ou concave (Fig. 1.17). Ainsi, une croissance de régions est appliquée en fonction de la catégorie assignée à chaque point.

Cependant, les raisons de l'utilisation de l'indicateur proposé ne sont pas claires. En effet, il est possible de discriminer les points correspondant aux arêtes directement à partir des courbures principales, souvent très élevées. De plus, la méthode repose sur de nombreux paramètres différents qui semblent difficiles à fixer pour un utilisateur non-expert, comprenant les voisinages de courbure pour les arêtes vives et lisses, ainsi que des tolérances de courbure pour discriminer les arêtes vives et chaque type de primitive recherché. Or, ces paramètres diffèrent en fonction du maillage.

$$s = -\frac{2}{\pi} \arctan \frac{1}{\sqrt{H^2 - K}}$$

where H and K are the Mean and Gaussian curvatures.

Regular and non-flat point classification based on the s index.

s value	Point classification
$s = -1$	Umbilical Concave (<i>UC</i>)
$-1 < s < -0.5$	Elliptical Concave (<i>EC</i>)
$s = -0.5$	Parabolic Concave (<i>PC</i>)
$-0.5 < s < 0$	Hyperbolic Concave (<i>HC</i>)
$s = 0$	Hyperbolic Saddle (<i>HS</i>)
$0 < s < 0.5$	Hyperbolic Convex (<i>HX</i>)
$s = 0.5$	Parabolic Convex (<i>PX</i>)
$0.5 < s < 1$	Elliptical Convex (<i>EX</i>)
$s = 1$	Umbilical Convex (<i>UX</i>)

FIGURE 1.17 – *Shape index* utilisé par Di Angelo et Di Stephano [30].

Synthèse

Les méthodes basées sur une croissance de régions sont souvent très simples à mettre en place, permettent un haut degré de liberté quant aux critères de regroupement, et enfin ne nécessitent pas de lourds calculs. Cependant, cette approche peut poser problème quant au choix de la graine, pouvant directement impacter les résultats. Mais des méthodes évitent ce problème en catégorisant au préalable les points ou triangles, permettant d'obtenir les mêmes régions avec des graines différentes.

Par fusion

Une troisième approche consiste à définir chaque élément comme une région indépendante, puis fusionner les régions adjacentes en fonction du critère de regroupement (Algorithme 3). Ainsi, l'algorithme se termine lorsqu'il n'est plus possible de fusionner deux régions voisines. Cette méthode permet de s'abstraire du problème du choix de la graine, mais nous pouvons alors nous confronter à un autre problème concernant l'ordre de fusion des régions. En effet, lorsqu'une région peut avoir plusieurs candidates pour la fusion, il est possible de la fusionner avec une seule, plusieurs, ou encore toutes les candidates en même temps. Chaque itération dépend donc des itérations précédentes, ce qui peut parfois produire des résultats très différents.

Algorithme 3 Fusion de régions sur les triangles

```

Créer un ensemble de régions R
Créer une nouvelle région pour chaque élément et l'ajouter à R
répéter
  pour tout région  $R_i$  de R faire
    pour tout région  $R_j$  de R différente de  $R_i$  faire
      si  $(R_i, R_j)$  peuvent être fusionnées alors
        Fusionner  $R_i$  et  $R_j$ 
      fin si
    fin pour
  fin pour
jusqu'à ce qu'il ne soit plus possible de fusionner des régions
renvoyer R

```

Kim *et al.* [53] proposent une méthode basée sur les angles dièdres (Fig. 1.18). Le point à noter dans leur méthode est leur manière de déterminer quelle paire de régions ils doivent fusionner en priorité. En effet, en choisissant de fusionner celles qui admettent l'angle dièdre le plus faible, les résultats ne dépendent plus de l'ordre des triangles. Ils proposent également de raffiner les résultats, notamment en améliorant les contours des régions.

Cependant, il semble difficile d'utiliser les angles dièdres pour des maillages contenant de nombreuses imprécisions. De plus, la complexité de cette méthode est suffisamment élevée pour largement impacter les temps de calculs pour des maillages très denses.

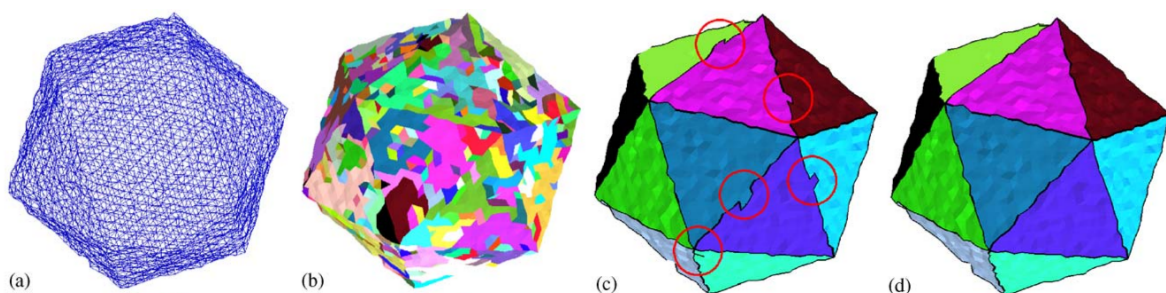


FIGURE 1.18 – Résultats de Kim *et al.* [53] : a) Maillage, b) Etat initial, c) Segmentation, d) Contours raffinés.

Garland *et al.* [39] proposent un algorithme basé sur un maillage dual et un critère de distance. Pour commencer, le maillage dual est construit : les barycentres des triangles sont calculés et reliés si les triangles correspondant sont voisins. Ainsi, chaque point du maillage dual correspond à une région, qui au départ ne contient qu'un seul triangle. Ensuite, la fusion de deux régions correspond à la contraction d'une arête du maillage dual et est réalisée si les deux régions sont proches d'un plan moyen, selon une tolérance de distance. Le processus est réitéré jusqu'à stabilité.

Cependant, le critère de distance utilisé est sensible à l'échelle des objets. De plus, les résultats obtenus montrent des régions essentiellement planes (Fig. 1.19). Les primitives géométriques incurvées comme les cylindres risquent alors d'être segmentées en plusieurs plans, comme dans la figure 1.19.c, ce qui peut complexifier leur extraction par la suite.

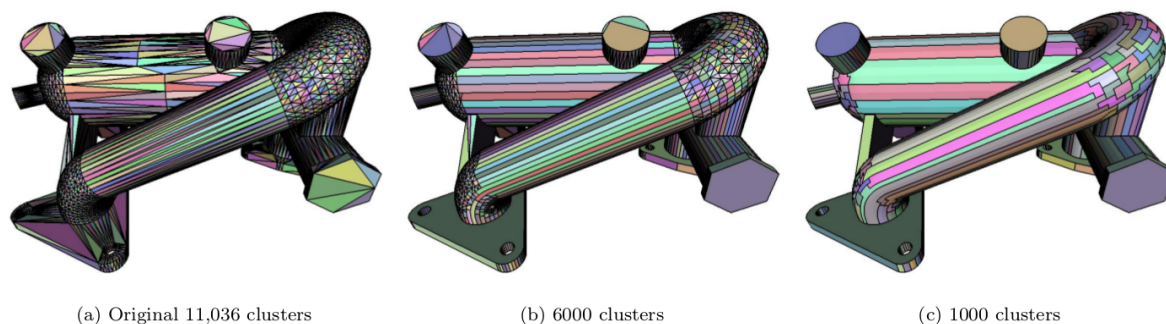


FIGURE 1.19 – Résultats de Garland *et al.* [39].

Dans la même idée, Attene *et al.* [2] présentent une méthode ayant pour objectif de déterminer des régions correspondant à des primitives. Ainsi, deux régions peuvent être fusionnées uniquement si l'ajustement d'une primitive sur les deux régions concaténées n'engendre pas une erreur trop grande, selon une tolérance de distance. Cette méthode est en quelque sorte une généralisation de celle de Garland *et al.* [39], en ajoutant d'autres types de primitives que le plan.

Cependant, de très nombreuses approximations de primitives sont réalisées à chaque itération, selon une complexité carrée du nombre de régions. De plus, il semble compliqué d'approximer une primitive géométrique correcte à partir d'une zone très restreinte d'un maillage imprécis, ce qui est notamment le cas pour les premières itérations.

Synthèse

Les méthodes basées sur une fusion de régions permettent de mieux maîtriser l'ordre de regroupement et ainsi mieux contrôler les résultats obtenus. Cependant, elles impliquent souvent des temps de calcul plus longs que pour les méthodes par croissance de régions. Il est également à noter que peu de méthodes récentes utilisent cette approche.

1.2.1.2 Par définition de frontières

Les zones homogènes d'un maillage peuvent parfois être définies à travers leurs contours. Dans la littérature, plusieurs méthodes recherchent des lignes caractéristiques dans les maillages 3D. Par exemple, Yoshizawa *et al.* [122] détectent les lignes de crête d'un objet à partir des valeurs de courbure. Ensuite, Watanabe et Belyaev [115] utilisent une surface focale, construite en translatant chaque point du maillage sur le centre de courbure correspondant. Les lignes de courbure saillantes sont alors déterminées en fonction des différences d'aires entre les triangles du maillage et leurs homologues sur la surface focale. Enfin, Gehre *et al.* [41] proposent une analyse des courbures à différents niveaux permettant de simplifier le maillage. Ces lignes caractéristiques peuvent ensuite être utilisées, par exemple, pour définir les contours des régions de l'objet. Il est à noter que les courbures sont souvent utilisées pour définir les frontières.

Lee *et al.* [60] proposent une segmentation basée sur une extraction de plans locaux (Fig. 1.20). Pour chaque paire de plans voisins, un segment de droite correspondant à leur intersection est calculé. Les segments ayant des extrémités proches, en distance euclidienne, sont alors reliés afin de créer des contours fermés. Enfin, les séquences de segments admettant une continuité relativement lisse permettent de définir des lignes caractéristiques. Cette méthode permet de séparer correctement les primitives, y compris les surfaces réglées.

Cependant, elle nécessite de paramétrer le nombre de régions sur lesquelles les plans sont approximés, comme pour les méthodes itératives. Cela pose notamment problème pour les objets complexes, et ne permet pas d'automatiser le processus sur des maillages divers.

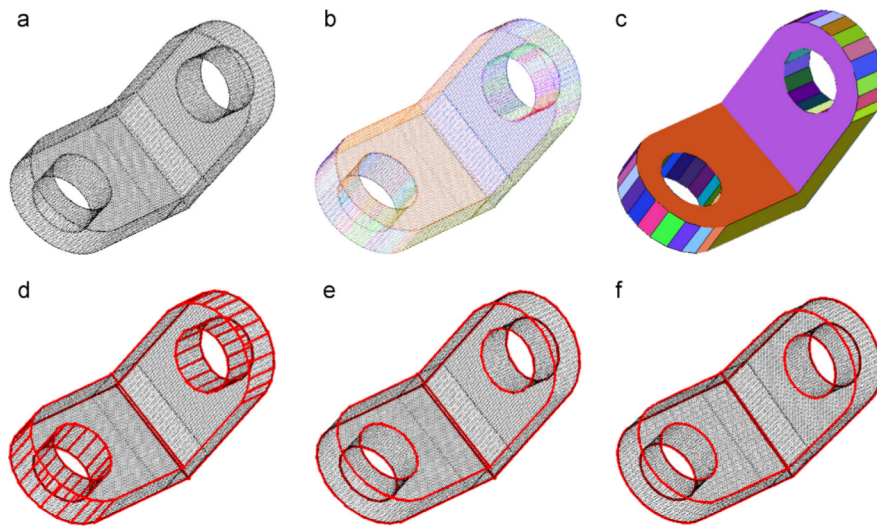


FIGURE 1.20 – Résultats de Lee *et al.* [60] : a-c) Approximation des plans, d-f) Construction des contours.

La méthode proposée par Várady *et al.* [109] est basée sur un complexe issu de la théorie de Morse, utilisant la courbure comme fonction de référence. La méthode commence par appliquer une segmentation par régions basée sur la courbure. Ensuite, une détection des arêtes est réalisée à l'aide d'un champ de vecteurs construit à partir des normales du maillage. Les arêtes sont alors regroupées en trois catégories : les arêtes vives, les arêtes lisses et les congés. Les arêtes vives, séparant les régions obtenues au préalable, et les arêtes lisses, situées à l'intérieur des régions, font directement partie des futurs contours. Les congés ayant une largeur non nulle sont quant à eux remplacés par deux courbes. Les contours obtenus sont ensuite simplifiés, lissés et recalés en fonction de la courbure, afin d'affiner les résultats.

Cependant, l'utilisateur doit définir un paramètre de sensibilité impactant fortement les résultats (Fig. 1.21). Ceci implique une certaine expertise ou de nombreux essais, afin d'utiliser correctement la méthode sur des maillages divers.

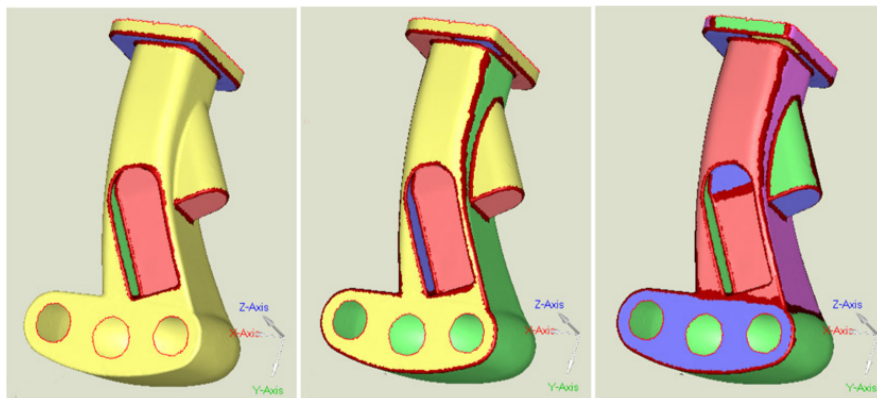


FIGURE 1.21 – Résultats de Várady *et al.* [109] avec une sensibilité croissante.

Demarsin *et al.* [28] proposent une extraction des arêtes vives à partir d'un nuage de points. L'objectif est de retrouver au final des contours fermés (Fig. 1.22). Dans un premier temps, le nuage est segmenté par croissance de régions sur les normales. Ensuite, un graphe est construit à partir du voisinage de ces régions (Fig. 1.22.a). De nouveaux arcs sont alors ajoutés au graphe, afin de fermer le contour qui contient souvent de nombreux *trous* à cause des imprécisions (Fig. 1.22.b). En effet, de nombreuses petites régions peuvent être présentes le long des arêtes vives de l'objet et partager leur voisinage avec des régions plus grandes. Enfin, le graphe est simplifié (Fig. 1.22.c et 1.22.d), fermé (Fig. 1.22.e) et lissé (Fig. 1.22.f).

Cependant, ces travaux sont très dépendants de la segmentation initiale, qui repose ici sur des normales sensibles aux imprécisions locales. Nous pouvons tout de même souligner l'importance de l'ajout de nouvelles arêtes dans le graphe, qui permet de compenser certaines erreurs au niveau des arêtes vives de l'objet.

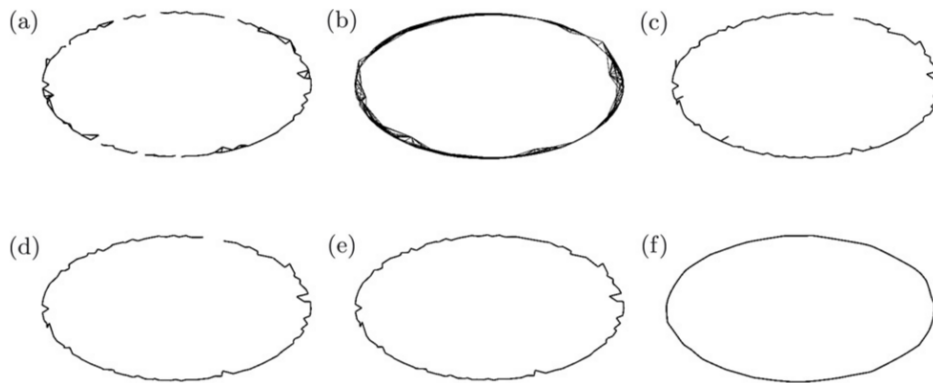


FIGURE 1.22 – Etapes de construction des contours fermés de Demarsin *et al.* [28].

Demarsin *et al.* [27] proposent aussi une méthode plus récente, basée sur la courbure. Pour commencer, un histogramme de la courbure moyenne non-signée est construit avec un nombre de classes prédéfini. Ensuite, l'histogramme est raffiné de sorte que chaque mode de l'histogramme soit représenté par une seule classe. Enfin, des seuils de segmentation sont définis à l'extrémité supérieure de chaque mode. Ces seuils permettent alors de discriminer les points internes aux régions de ceux des frontières. Ces derniers sont ensuite utilisés pour reconstruire des contours fermés, à partir des régions correspondantes.

Cependant, le nombre de classes de l'histogramme a un fort impact sur la détermination des seuils (Fig. 1.23). Or, puisque l'histogramme est construit entre les valeurs extrêmes de courbure et que celles-ci dépendent à la fois de la forme et de l'échelle de l'objet, il n'est pas possible de déterminer une résolution unique pour tous les maillages. Ainsi, les auteurs proposent une analyse multi-résolutions de cet impact et fixent le nombre de classes optimal à 50. Malheureusement, cette analyse est faite à partir d'un seul maillage, et sa généralisation reste discutable. Enfin, l'utilisateur doit à deux reprises valider ou modifier des paramètres, ce qui n'en fait pas une méthode automatique.

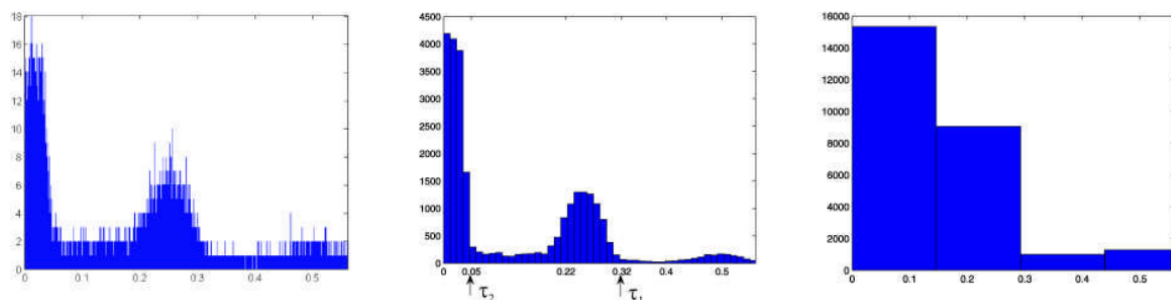


FIGURE 1.23 – Histogramme de courbure moyenne du même maillage avec 30 000, 50 et 4 classes [27].

Chen *et al.* [21] proposent également d'analyser un histogramme de courbure afin de définir des seuils de segmentation. Pour commencer, un histogramme de courbure moyenne signée est construit. En effet, le signe de la courbure permet de distinguer les zones convexes et concaves du maillage. Ensuite, un lissage Laplacien modifié est appliqué à l'historgramme afin de compenser les chevauchements dûs aux imprécisions des données. L'historgramme présente alors des modes distincts, permettant de définir des seuils de segmentation les séparant. Ainsi, chaque intervalle de courbure moyenne entre deux seuils permet de définir une zone homogène, correspondant souvent à une ou plusieurs primitives géométriques (Fig. 1.24). Les valeurs extrêmes restantes correspondent aux arêtes vives de l'objet. Enfin, les contours sont reconstruits et corrigés afin d'obtenir des bandes de triangles les plus fines possibles.

Cependant, la courbure dépendant de l'échelle de l'objet et l'historgramme n'étant pas normalisé, cette approche ne permet pas de traiter tous les maillages de la même manière. De plus, le lissage dépend de paramètres experts et ne permet pas de séparer des intervalles de courbure trop proches. Notons que cet histogramme est construit en définissant la largeur des classes au lieu de leur nombre. Cela permet une meilleure adaptation aux plages variables des valeurs de courbure. Cependant, cela ne règle pas le problème d'échelle nécessitant de modifier cette largeur.

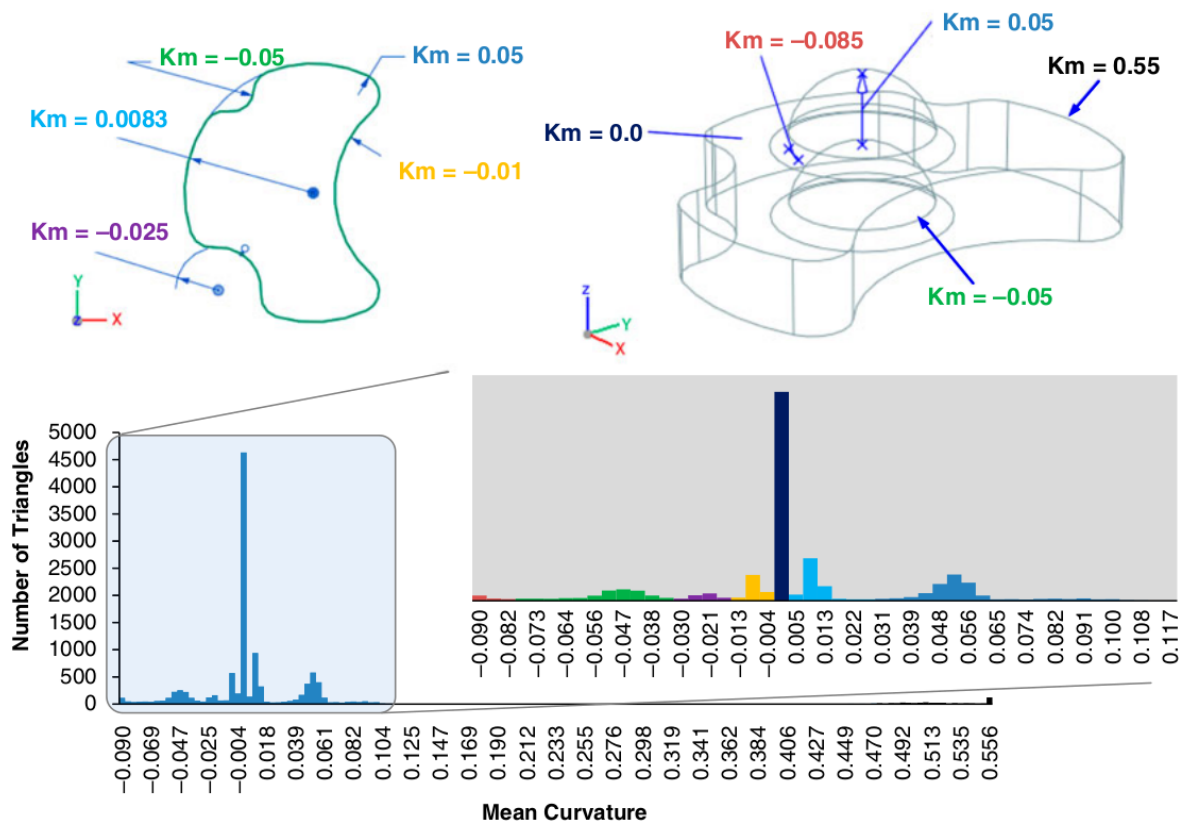


FIGURE 1.24 – Résultats de Chen *et al.* [21], montrant la correspondance entre courbures et primitives.

Dans une approche plus générale, Benhabile *et al.* [4] proposent une segmentation basée sur de l'apprentissage. Une base de données est construite au préalable avec de nombreuses segmentations manuelles de plusieurs objets. Ensuite, les contours produits par les différents utilisateurs sont analysés afin de déterminer des tendances, en calculant et comparant plusieurs descripteurs tels que les angles dièdres ou les courbures. Des contours candidats sont alors automatiquement proposés sur un nouvel objet, en fonction des caractéristiques communes avec ceux de la base de données. Ces contours candidats sont ensuite raffinés afin d'obtenir des contours fermés. Cette approche permet d'être efficace sur la plupart des maillages.

Cependant, une base de données très riche et de nombreux utilisateurs sont nécessaires avant de pouvoir traiter correctement des maillages inconnus.

Au premier abord, il semblerait plus pertinent de procéder par frontières puisque les primitives contenues dans un objet manufacturé sont souvent délimitées par des arêtes très marquées. Cependant, cette approche est souvent basée sur une première segmentation par régions, et semble parfois compliquée dans le cas de maillages issus de numérisation. En effet, les scanners actuels impliquant des imprécisions, les frontières ne sont pas clairement définies et correspondent plutôt à des régions semblables à de petits congés. Au final, il semble que rechercher directement les régions semble plus adapté à notre problématique.

Parmi ces approches, il semble que la croissance de régions soit le meilleur algorithme candidat pour concevoir une méthode rapide et automatique. Nous remarquons également que de nombreuses méthodes de segmentation se basent sur les courbures, les primitives géométriques étant souvent extraites par la suite à l'aide de celles-ci. Mais ces approches ne sont souvent ni assez robustes aux imprécisions, ni automatiques. En effet, les valeurs de courbure dépendent de plusieurs paramètres très variables, dont la forme, l'échelle et la densité du maillage par exemple. Il faudrait donc concevoir une méthode s'adaptant automatiquement en fonction des caractéristiques du maillage.

1.2.2 Caractérisation locale de la forme 3D par courbure

Pour définir la forme locale d'un maillage, des indicateurs de forme comme les normales ou les courbures sont souvent calculées. A partir d'un maillage 3D, le calcul des normales est bien maîtrisé [105] mais n'est pas suffisant pour caractériser efficacement la forme des maillages issus de numérisation. En effet, cette approximation de premier ordre ne permet que de définir un plan localement. La courbure, en revanche, est une approximation de second ordre définissant une quadrique. Elle permet ainsi de mieux définir la forme locale et de manière plus robuste.

1.2.2.1 Calcul des courbures

Intuitivement, le calcul des courbures permet de quantifier la déviation entre une courbe et une droite en 2D, ou entre une surface et un plan en 3D. La valeur de courbure d'une courbe 2D à un point P est égale à l'inverse du rayon du cercle osculateur en P. Ce cercle osculateur correspond à l'arc circulaire approximant au mieux la courbe au voisinage de P (Fig. 1.25.a).

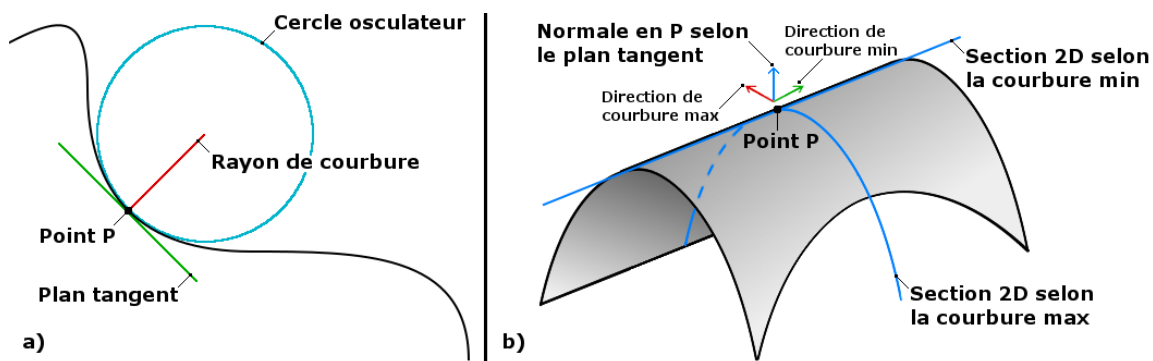


FIGURE 1.25 – Représentation de la courbure sur : a) Une courbe 2D, b) Une surface 3D.

Sur une surface 3D, une infinité de directions de courbure existent autour de la normale de P (Fig. 1.25.b), suivant des plans sectionnant la surface. Dans ce cas, il est nécessaire de définir certaines courbures particulières. Ainsi, les courbures principales correspondent aux courbures minimale et maximale, dont les directions sont perpendiculaires. De même, nous pouvons définir les courbures moyenne et Gaussienne, correspondant respectivement à la moyenne et au produit des courbures principales. Plus généralement, la formule d'Euler définit la courbure continue en un point P pour chaque vecteur tangent \vec{t}_i telle que :

$$k_n(t_i) = k_{max} \cos^2(\theta) + k_{min} \sin^2(\theta), \quad (1.1)$$

où k_{max} et k_{min} sont les courbures principales, et θ est l'angle entre la direction de courbure maximale \vec{d}_{max} et le vecteur tangent \vec{t}_i .

Sachant qu'un maillage est un objet discret, la courbure doit être approximée à partir d'un ensemble fini de points. Pour cela, de nombreuses méthodes se sont succédées dans la littérature. Gatzke et Grimm [40] présentent un état de l'art sur le calcul des courbures en définissant trois catégories de méthodes, basées sur des quadriques, des angles dièdres ou des tenseurs de courbure. Magid *et al.* [67] proposent une comparaison de ces méthodes en fonction des caractéristiques du maillage. Cette dernière suggère que les méthodes existantes ne permettent pas de traiter efficacement tous les types de maillages.

Pour répondre à cette problématique, Bénéière *et al.* [5] proposent une méthode de calcul de courbure basée sur les variations des normales. Ainsi, pour chaque voisin de P, une valeur de courbure discrète est calculée avec la formule de Dong et Wang [32] :

$$k_n(t_i) = \frac{\overrightarrow{PP_i} \cdot \overrightarrow{NN_i}}{\|\overrightarrow{PP_i}\|^2}, \quad (1.2)$$

où P_i est un point voisin de P ayant une normale \vec{N}_i , et \vec{t}_i la projection de $\overrightarrow{PP_i}$ sur le plan tangent à la surface en P.

Ensuite, les courbures principales en P sont déterminées à l'aide des régressions proposées par Chen et Schmitt [24]. Celles-ci permettent d'obtenir une meilleure précision qu'en choisissant parmi les courbures discrètes calculées, tout en étant plus robuste aux imprécisions présentes dans les maillages 3D issus de numérisation.

1.2.2.2 Analyse de la distribution des courbures

Comme de nombreuses méthodes de rétro-ingénierie, celle proposée par Bénéière *et al.* [8] analyse uniquement les valeurs de courbures de manière locale pour extraire les primitives géométriques. Cependant, il est parfois nécessaire de retrouver des informations plus globales. En effet, les maillages issus de numérisation contiennent de nombreuses imprécisions pouvant altérer les analyses locales. Ainsi, plusieurs méthodes construisent la distribution des courbures sur tout le maillage 3D afin d'analyser ce dernier dans sa globalité [1, 50].

Nous pouvons dans un premier temps distinguer les distributions discrètes et continues. Une distribution discrète est souvent représentée par un histogramme (Fig. 1.26.a). D'un autre côté, une distribution continue correspond à un modèle mathématique. Les distributions Gaussiennes et les lois Normales sont les plus fréquemment utilisées (Fig. 1.26.b). Les distributions statistiques sont par ailleurs utilisées dans de nombreux autres domaines, comme la compression d'images [69], les systèmes de hash [45], ou encore pour automatiser certains traitements d'images [49].

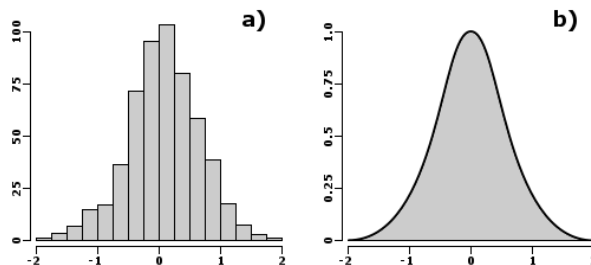


FIGURE 1.26 – Exemples de distributions : a) Empirique, b) Loi de probabilité.

En particulier, les distributions de courbures permettent de caractériser la forme d'un maillage. Tout d'abord, le signe des courbures permet de discriminer les zones convexes et concaves. En effet, une région convexe donne des valeurs négatives et une région concave des valeurs positives. Dans le cas d'une zone plane, les valeurs de courbure sont nulles. Cela permet alors de caractériser la forme locale, soit pour chaque point indépendamment, soit pour une région à partir d'une

distribution. Il est ainsi possible d'en déduire un type de primitive géométrique en fonction du signe des courbures principales (Fig. 1.27).

Ainsi, une zone plane correspond à un ensemble de points ayant des valeurs de courbures nulles. De même, une région sphérique admet des courbures minimale et maximale similaires mais non-nulles. Notons que l'inverse de la courbure correspond dans ce cas au rayon de la sphère. Pour le cylindre, et de manière plus générale le cône, une des deux courbures principales admet des valeurs nulles, et l'autre des valeurs non-nulles. Enfin, cette caractérisation de la forme peut permettre de guider un processus de segmentation ou d'ajustement par exemple.

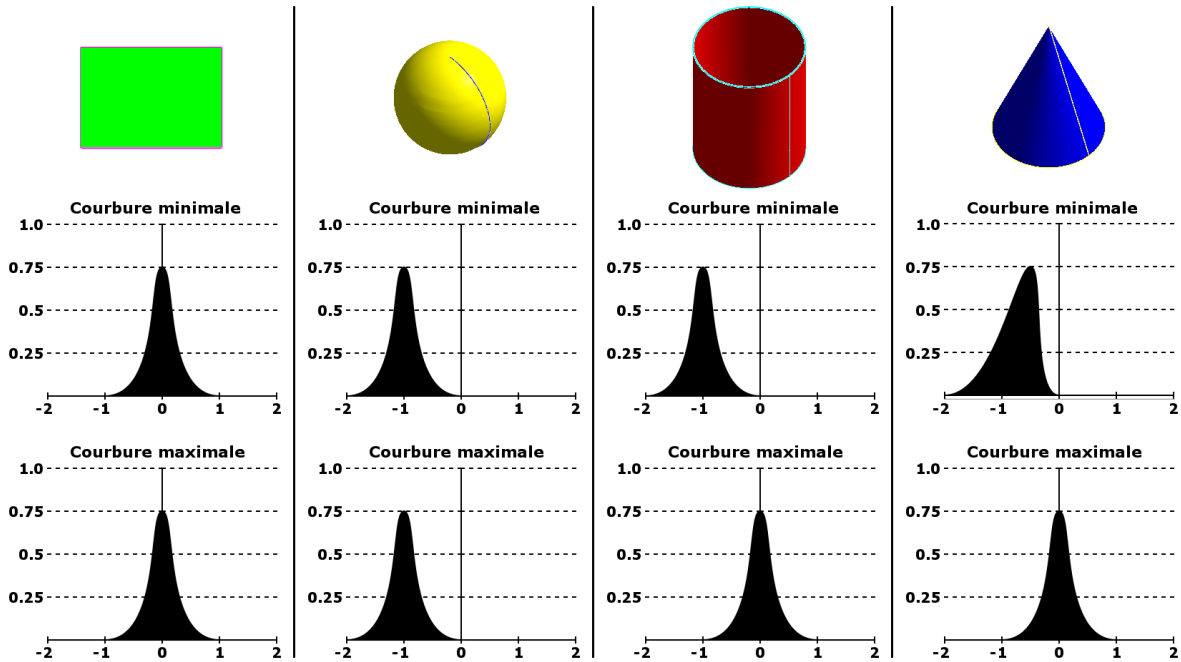


FIGURE 1.27 – Caractérisation de la forme grâce aux distributions des courbures principales.

Dans la section 1.2.1, nous avons vu plusieurs méthodes de segmentation basées sur un histogramme de courbure moyenne : non-signée pour Demarsin *et al.* [27] et signée pour Chen et Feng [21]. Ces travaux montrent qu'il est possible de déterminer une ou plusieurs valeurs de seuil de courbure, permettant alors de séparer la distribution en intervalles de courbures homogènes. Ces intervalles sont principalement détectés par une recherche des pics et des vallées dans l'histogramme. Ils permettent au final de séparer les régions géométriquement différentes, correspondant par exemple à des primitives géométriques.

Cependant, il est nécessaire de construire un histogramme *stable* et représentatif afin de calculer les bons seuils de courbure. Or, un histogramme classique n'est pas adapté à l'échelle des valeurs de courbure qui est infinie, posant problème pour le choix du nombre de classes. En effet, la forme de l'histogramme peut fluctuer énormément selon le pas de discrétisation, en fonction de la répartition des valeurs au sein de chaque classe (Fig. 1.28). Les seuils calculés peuvent donc être très différents en fonction de la construction de l'histogramme.

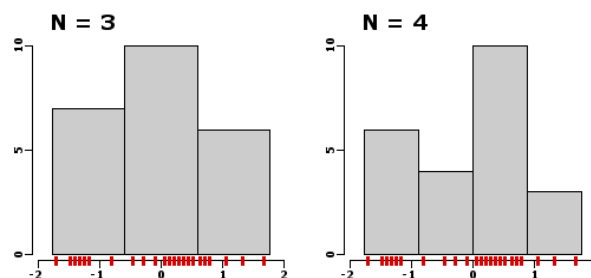


FIGURE 1.28 – Problème de l'histogramme : la forme change selon la répartition des valeurs sur l'axe.

Une approche classique consiste à approximer une distribution à l'aide d'un modèle continu. McLachlan et Krishnan [71] présentent l'algorithme *Expectation-Maximisation* (EM). Ils commencent par définir un modèle continu, par exemple Gaussien. Ensuite, l'algorithme réalise itérativement les deux étapes suivantes :

- Evaluation de l'espérance (E) : calculer un score de vraisemblance entre le modèle et les données mesurées;
- Maximisation (M) : estimer de meilleurs paramètres pour le modèle en maximisant la vraisemblance trouvée à l'étape E.

Les paramètres du modèle obtenus dans l'étape M sont ensuite utilisés comme point de départ d'une nouvelle étape d'évaluation. Le processus est réitéré jusqu'à obtenir un score acceptable ou un modèle stable. Cependant, un histogramme ne contient que rarement un seul mode.

Une première alternative consiste à définir plusieurs modèles afin de mieux représenter la distribution. Ainsi, Masmoudi *et al.* [68] présentent un algorithme de mélange de modèles. Celui-ci peut être vu comme la fusion entre les algorithmes *K-Means* et *Expectation-Maximisation*. Il commence par définir N modèles, dont les paramètres peuvent être aléatoirement choisis, puis se déroule en trois étapes :

- Pour chaque mesure, calculer un score de vraisemblance pour chaque modèle et associer la mesure au modèle obtenant le meilleur score;
- Evaluer l'espérance de chaque modèle en fonction des mesures associées;
- Maximiser la vraisemblance pour chaque modèle et déterminer de meilleurs paramètres.

Comme pour l'algorithme EM, l'ensemble des paramètres de chaque modèle calculés à la troisième étape constitue le point de départ d'une nouvelle itération. Le processus est réitéré jusqu'à stabilité de tous les modèles. Cette méthode permet de représenter correctement la plupart des distributions statistiques (Fig 1.29).

Cependant, celle-ci nécessite de connaître au préalable le nombre de modèles permettant de définir au mieux la distribution. En effet, des techniques comme le calcul du critère d'information bayésien [94] permettent de proposer un nombre de modèles, mais ont tout de même leurs limites. De plus, la complexité importante de l'algorithme peut engendrer des temps de calcul importants. Enfin, le modèle utilisé impacte directement la représentation finale de la distribution. Or, ce modèle n'est pas toujours simple à déterminer.

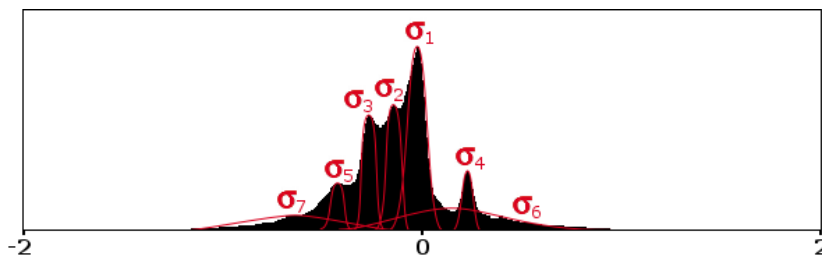


FIGURE 1.29 – Exemple de distribution approximée par un mélange de modèles Gaussiens.

Une seconde alternative est de restreindre la portée des modèles. Carreira-Perpiñán [20] présente un état de l'art sur l'algorithme du *Mean-Shift*. Celui-ci est une technique d'analyse statistique permettant de localiser les modes d'une distribution de données mesurées (Fig. 1.30). Comme pour l'approche précédente, il est nécessaire de définir une fonction de densité. Ensuite, les modèles sont mis à jour en fonction d'une zone de recherche définie en fonction d'un paramètre de largeur h . La principale différence avec l'approche précédente est donc de donner la largeur h de la fenêtre de recherche au lieu du nombre de modes recherchés. En effet, il semble que cette largeur puisse avoir plus de sens, selon les applications visées, que le nombre de modes.

Cependant, la largeur de la fenêtre n'a pas toujours un sens particulier, notamment pour une distribution de courbure. De plus, comme pour la majorité des méthodes d'optimisation, les temps de calculs ne sont pas linéaires.

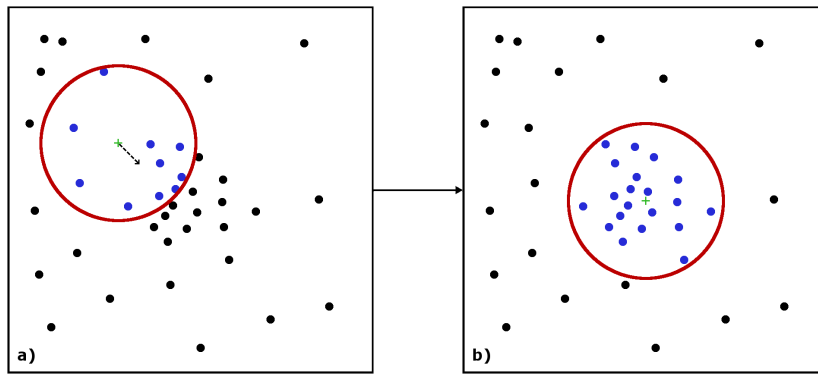


FIGURE 1.30 – Exemple d'utilisation du Mean-Shift : a) Etat initial, b) Etat final.

Une seconde approche consiste à réaliser une estimation par noyau (Fig. 1.31). Parzen [84] présente ainsi un algorithme prenant un unique modèle comme paramètre. Prenons le cas d'une distribution de courbure approximée à l'aide d'un modèle Gaussien. La première étape consiste à définir l'écart-type du modèle Gaussien, qui reste fixe. Ensuite, un modèle Gaussien est défini pour chaque valeur de courbure, en considérant cette dernière comme sa moyenne. Enfin, tous les modèles Gaussiens sont ajoutés pour obtenir la distribution globale. Cette approche est en outre robuste aux valeurs aberrantes, qui impactent très peu la distribution finale.

Cependant, cette approche nécessite de définir un modèle précis. Or, une distribution de courbure implique une plage de valeurs potentiellement infinie. Dans ce cas, il semble compliqué de déterminer correctement des paramètres tels que l'écart-type utilisé pour estimer la distribution avec un modèle Gaussien.

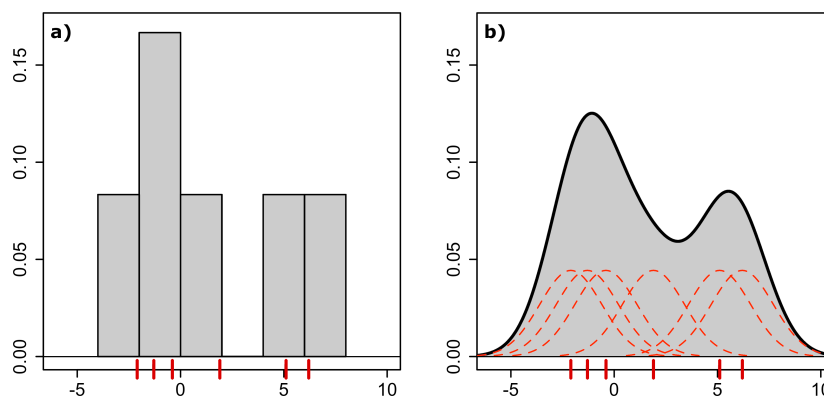


FIGURE 1.31 – Exemple de distribution estimée par noyau Gaussien : a) Histogramme, b) Estimation.

Synthèse

Les distributions de courbures permettent de faire ressortir des informations globales. Elles permettent par exemple de discriminer les points en fonction de la primitive associée. Cependant, l'utilisation d'un histogramme est complexe car adaptée ni aux valeurs de courbures qui sont des réels, ni à l'échelle des objets impliquant une plage de valeurs variable.

Les méthodes d'approximation de distributions continues semblent plus précises. Cependant, ces approches sont paramétrées, ce qui implique souvent un manque d'automatisme. La dernière approche d'estimation par noyau semble être un bon compromis entre la précision, la robustesse et les temps de calculs. Il faut cependant définir précisément le modèle utilisé, de manière à traiter des distributions très diverses, ce qui n'est pas trivial.

1.2.3 Ajustement d'une primitive sur un nuage de points

L'extraction de primitives a pour objectif de reconstruire des surfaces continues à partir d'un ensemble fini de points 3D (Fig 1.32). Nous cherchons généralement à extraire des primitives simples telles que des plans, sphères et cylindres car ce sont les plus utilisées dans les objets manufacturés. Mais nous pouvons également rechercher d'autres primitives, telles que des cônes, des tores, ou encore des surfaces réglées par exemple. Il existe plusieurs approches permettant d'estimer les paramètres des primitives géométriques à partir d'un nuage de points ou d'un maillage.

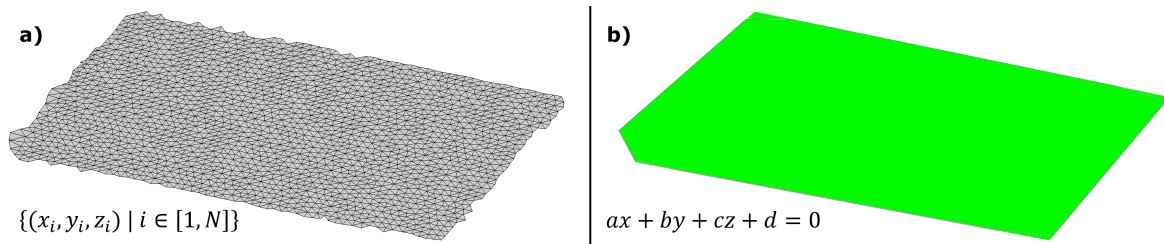


FIGURE 1.32 – Ajustement d'une primitive : a) Points 3D mesurés, b) Surface approximée.

Par profil 2D

L'ajustement proposé par Benkő *et al.* [10] est basé sur la reconstruction d'un profil 2D. Ainsi, les points du maillage sont tout d'abord projetés sur un plan. Ensuite, des segments de droite et arcs de cercle sont extraits du nuage de points 2D obtenus. Des contraintes de continuité sont alors appliquées sur ces segments de droite et arcs de cercle afin de construire un profil continu. Enfin, les surfaces sont reconstruites par extrusion ou révolution.

Cependant, l'orientation choisie pour le plan impacte beaucoup le profil reconstruit. En effet, la distinction entre extrusion et révolution réside principalement dans l'orientation de ce plan. Par exemple, le profil d'extrusion d'un cylindre représente un cercle alors que son profil de révolution est une droite. Or, il semble difficile de choisir cette orientation automatiquement, notamment pour des maillages issus de numérisation contenant de nombreuses imprécisions.

Par étude de la sphère Gaussienne

Une autre approche consiste à analyser la distribution des normales correspondant aux points. Pour cela, nous pouvons étudier une sphère Gaussienne. Cette dernière est une sphère unitaire dans laquelle sont définis des vecteurs unitaires représentés par des points. Dans notre cas, ces vecteurs sont définis par les normales des points du maillage.

Toony *et al.* [106] proposent ainsi de construire une sphère Gaussienne pour caractériser le type de primitive (Fig. 1.33). En effet, les normales correspondant à un plan sont toutes similaires (Fig. 1.33.c). Cela implique que tous les points de la sphère Gaussienne correspondante soient égaux (Fig. 1.33.h), ou plutôt forment une *tache* dans le cas de maillages imprécis. D'un autre côté, un cylindre implique une distribution des normales le long d'un cercle, qui lui-même est centré par rapport à la sphère (Fig. 1.33.g). De même, un cône engendre un cercle, mais cette fois excentré en fonction de son demi-angle (Fig. 1.33.f).

Cependant, plus les primitives deviennent complexes, plus il est difficile de les distinguer. En effet, la distribution des normales d'une surface complexe risque de couvrir toute la sphère sans produire de motif particulier (Fig. 1.33.j). Ainsi, les auteurs proposent d'utiliser un accumulateur Gaussien afin de différencier par exemple la sphère du tore, mais la méthode semble difficilement extensible à des surfaces plus complexes.

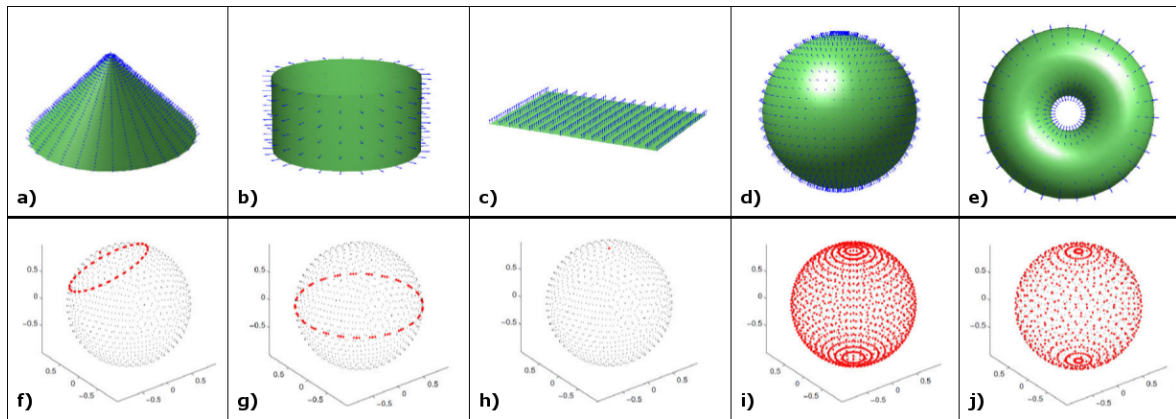
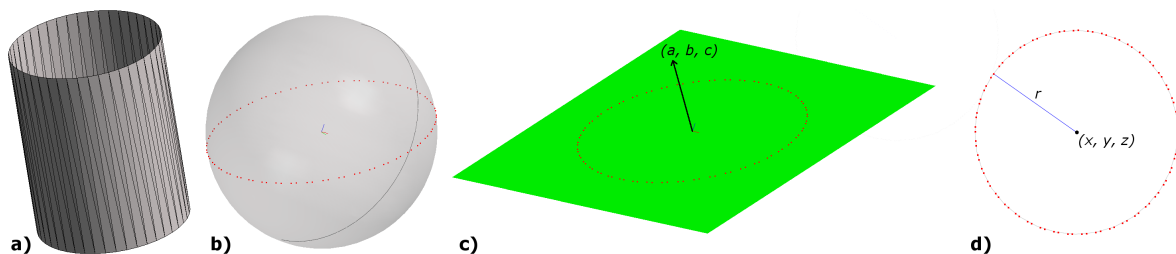


FIGURE 1.33 – Normales associées à des primitives [106] : a-e) Sur le maillage, f-j) Sur la sphère Gaussienne.

Par régression

Fleishman *et al.* [36] proposent une méthode basée sur des régressions linéaires. Ainsi, un plan est ajusté localement sur les points, afin de déterminer si ces points correspondent à une surface plane ou courbée. Un processus de sélection des points aberrants est également appliqué afin de choisir le meilleur sous-ensemble de points. Grâce à cela, les auteurs définissent une projection des points sur des surfaces, permettant une meilleure reconstruction. Cependant, le plan est le seul type de primitive que l'on peut approximer par régression linéaire. En effet, l'ajustement d'une sphère, d'un cylindre ou d'un cône, par exemple, n'est pas linéaire à cause du calcul des distances correspondantes.

Ainsi, Bénéière *et al.* [6] proposent d'utiliser une composition de régressions pour les cas non linéaires. En effet, des heuristiques permettent de déterminer successivement les paramètres de ces primitives. Pour le cylindre par exemple, l'orientation de l'axe est déterminée à partir d'une sphère Gaussienne, qui contient toutes les normales des points (Fig. 1.34.b). Les points de la sphère Gaussienne forment alors un plan, approximé par régression linéaire, dont la normale correspond à l'orientation de l'axe recherché (Fig. 1.34.c). Ensuite, les points du maillage peuvent être projetés sur le plan, décrivant alors un cercle 2D. Ce dernier peut alors être approximé par régression, son centre donnant la position de l'axe et son rayon correspondant à celui du cylindre (Fig. 1.34.d). Une alternative consiste à calculer la courbure, puis approximer l'ensemble des centres de courbure par une droite, formant alors l'axe. Ensuite, le rayon est calculé à partir des distances entre cet axe et les points.


 FIGURE 1.34 – Ajustement par combinaison de régressions de Bénéière *et al.* [6] : a) Maillage, b) Sphère Gaussienne, c) Plan approximé par régression, d) Cercle approximé par régression.

Cependant, la combinaison de régressions ne permet pas d'assurer l'optimalité des paramètres dans leur ensemble. En effet, chaque paramètre est alors dépendant des précédents, entraînant une propagation des erreurs. Par exemple, si l'axe d'un cône est mal approximé à cause des imprécisions sur les données, le sommet obtenu, étant contraint à être sur l'axe, peut souffrir d'une imprécision encore plus grande.

Par optimisation non-linéaire

Pour approximer les primitives par optimisation non-linéaire, de nombreux algorithmes existent. Un des plus fréquemment utilisés est celui de Levenberg-Marquardt [76], qui permet en outre de ne pas tomber dans un minimum local. L'objectif est, à partir d'un ensemble de points 3D et de paramètres initiaux de la primitive recherchée, de minimiser la somme quadratique des distances entre les points et la surface. Shakarji [96] propose d'ailleurs des fonctions de distances et leurs dérivées, entre un point et une primitive géométrique. Ces dérivées sont alors estimées dans le processus d'optimisation afin de calculer un gradient. Ainsi, ces méthodes permettent de mieux estimer conjointement tous les paramètres des primitives recherchées.

Cependant, cette approche nécessite souvent une initialisation correcte des paramètres de la primitive recherchée afin de pouvoir converger. Malheureusement, dans le cas d'une extraction de primitive, nous pouvons *a priori* rarement connaître les paramètres au début du processus. De plus, les calculs nécessaires à l'approximation d'une primitive peuvent devenir très lourds à cause des distances calculées à chaque itération. Ces méthodes ne peuvent donc pas être utilisées seules. Il serait alors intéressant, par exemple, d'initialiser une étape d'optimisation à l'aide des paramètres issus d'une combinaison de régressions.

Par contrainte

Enfin, une autre technique consiste à ajuster des primitives en imposant des contraintes géométriques. Benkő *et al.* [9] proposent ainsi de définir une liste de contraintes géométriques, comme par exemple le parallélisme ou la concentricité, selon un ordre de priorité. Ensuite, les paramètres des primitives sont raffinés itérativement en prenant en compte ces contraintes. Celles-ci sont appliquées séquentiellement, de sorte que chaque nouvelle contrainte imposée soit cohérente avec les précédentes. Ainsi, de nombreuses itérations permettent de régulariser le système (Fig. 1.35).

Cependant, cet ajustement peut impliquer des temps de calcul importants, dépendant des quantités de points, de primitives et de contraintes. De plus, les contraintes imposées doivent être déterminées au préalable.

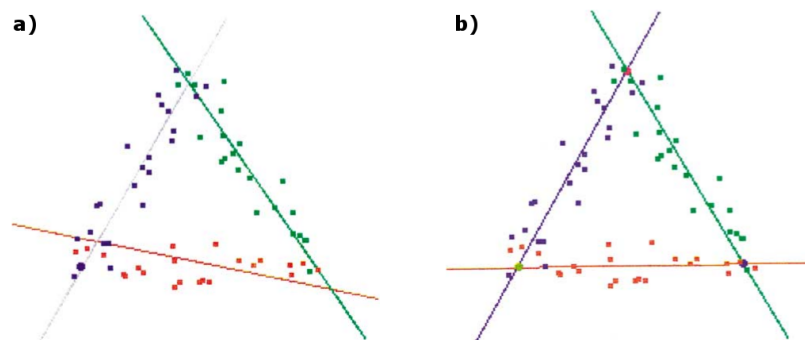


FIGURE 1.35 – Ajustement par contraintes de Benkő *et al.* [9] : a) Etat initial ne respectant aucune contrainte, b) Etat final respectant des angles de 60° entre les droites.

Synthèse

Parmi les approches d'ajustement de primitives, celles par profil 2D ou par étude de la sphère Gaussienne semblent trop spécifiques à certains types de primitives. Cela limite malheureusement l'extensibilité de ces méthodes. Parmi les autres approches, la moins contraignante et la plus rapide semble être la régression, ou plus particulièrement la combinaison de régressions. Cependant, cette dernière peut souffrir d'imprécisions pour les cas non linéaires puisque tous les paramètres ne sont pas calculés en même temps, créant de ce fait une propagation d'erreur. D'un autre côté, l'approche la plus précise semble être l'optimisation non-linéaire. Cependant, elle nécessite une bonne initialisation pour converger et peut être longue à calculer. Afin de concevoir une méthode rapide et précise, il semble donc nécessaire d'utiliser conjointement les méthodes de régression et d'optimisation.

1.3 Méthodes directes

Les méthodes directes ont pour objectif d'extraire des primitives directement ou globalement à partir du maillage 3D, sans segmentation préalable. La plupart du temps, ces méthodes n'ont pas besoin d'informations sur le maillage concernant la topologie de l'objet ou l'orientation des normales. Elles sont ainsi particulièrement adaptées aux nuages de points.

Par sections 2D

Une première approche consiste à sectionner l'objet, selon une ou plusieurs directions, pour ensuite reconstruire le modèle à partir des coupes 2D [112]. Cela permet d'extraire les primitives de manière très générique. Comme nous avons pu le voir avec les travaux de Benkó *et al.* [10], cette approche est particulièrement adaptée pour les extrusions et les surfaces de révolution (Fig. 1.36).

Cependant, il peut être difficile de définir l'orientation des coupes à effectuer, ainsi que leur nombre et l'espace entre deux sections. Ces trois paramètres peuvent en effet être susceptibles de changer localement si l'on veut reconstruire le modèle de manière optimale. On remarque par exemple que l'objet de la figure 1.36.d ne contient pas les deux encoches de celui de la figure 1.36.a. De même, en choisissant une section traversant ces encoches, seul le plateau inférieur est reconstruit. Les méthodes utilisant cette approche ne sont donc pas automatiques et nécessitent un utilisateur expert pour guider la construction des sections.

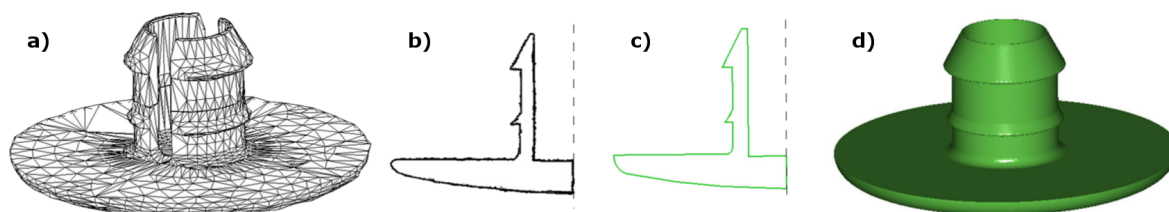


FIGURE 1.36 – Résultats de Wang *et al.* [112] : a) Maillage 3D, b) Section 2D par un plan contenant l'axe de révolution, c) Contour reconstruit, d) Surface de révolution reconstruite.

Par propagation locale

Une seconde approche consiste à procéder par propagation des primitives sur les points ou les triangles, à partir de graines, à l'image d'une croissance de régions.

Par exemple, la méthode proposée par Oesau *et al.* [80] commence par la construction d'un Octree, permettant de définir une relation de proximité entre les points 3D. Ensuite, des plans sont calculés à partir de sous-ensembles de points proches. Ces plans locaux sont alors étendus à d'autres points puis approximés à nouveau en fonction du nouveau sous-ensemble formé. Les plans sont également régularisés les uns par rapport aux autres, en fonction d'un graphe de relations parallèles et orthogonales. L'idée principale est de guider l'extraction des plans afin d'obtenir un modèle robuste aux imprécisions (Fig. 1.37).

Malheureusement, la méthode n'est définie que pour les plans, ce qui limite ses applications. Notons tout de même que les itérations effectuées permettent d'augmenter la quantité de points associés à chaque plan tout en éliminant ceux qui s'en éloignent.

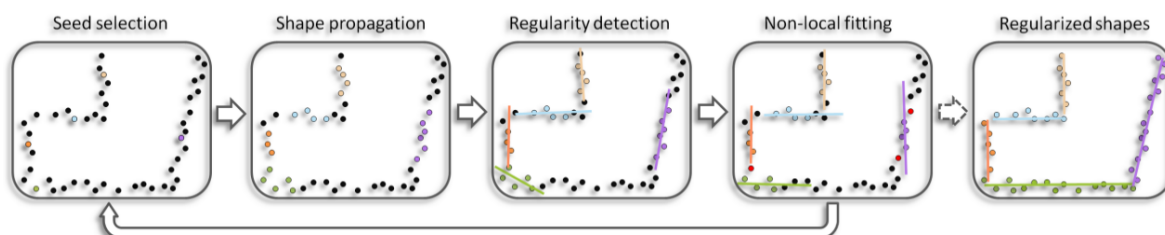


FIGURE 1.37 – Schéma global de la méthode de Oesau *et al.* [80].

Tran *et al.* [107] proposent une méthode plus générale, permettant d'extraire des cylindres, sphères et plans, localement dans un nuage de points (Fig. 1.38). Celle-ci commence par choisir un point au hasard, récupère ses points voisins et approxime toutes les primitives recherchées sur ce sous-ensemble. Enfin, elle ne conserve que la primitive ayant l'erreur la plus petite. Il est également important de noter que l'ordre des primitives approximées est important. En effet, les auteurs choisissent volontairement de rechercher d'abord les surfaces courbes (cylindres et sphères), puis les plans dans une étape à part. Cela permet de ne pas confondre, par exemple, des portions de sphères ou de cylindres avec des plans.

Cependant, ce dernier point laisse à penser que l'extraction directe n'est pas toujours robuste aux imprécisions et que la méthode ne permet pas de retrouver un plan si les points correspondants ont déjà été approximés par une sphère de très grand rayon par exemple.

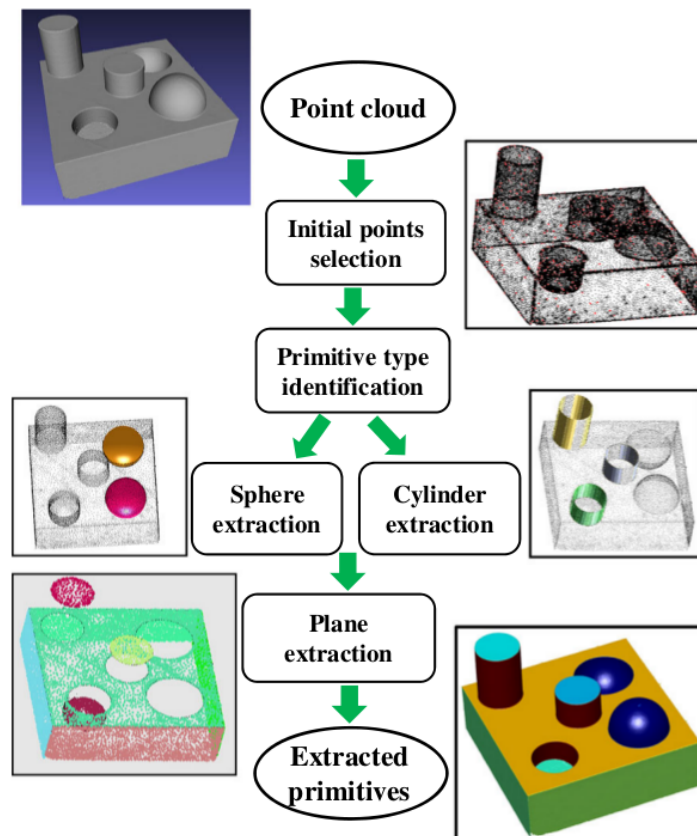


FIGURE 1.38 – Schéma global de la méthode de Tran *et al.* [107].

Par sélection de sous-ensembles de points

L'algorithme RANSAC (RANdom SAMple Consensus) [34] est une des premières méthodes directes, mais aussi une des plus utilisées. Cet algorithme commence par sélectionner un sous-ensemble de N points aléatoirement, à partir duquel une primitive est approximée. Ensuite, tous les points proches de la primitive, selon une tolérance de distance, sont ajoutés au sous-ensemble courant. Enfin, la primitive est conservée ou non en fonction du nombre de points du sous-ensemble final et d'un paramètre d'échelle. L'algorithme est ensuite itéré afin de maximiser la taille des sous-ensembles. Les points forts de cette méthode sont sa capacité à traiter des nuages de points et une très bonne robustesse aux imprécisions sur les données (Fig. 1.39).

Cependant, elle est très coûteuse en temps, principalement à cause des nombreuses itérations nécessaires pour aboutir à une solution satisfaisante. En effet, la combinatoire du processus de recherche d'une primitive peut parfois être très importante, par exemple lorsque l'on cherche un plan nécessitant 3 points au minimum dans un nuage de plusieurs millions de points. De plus, cet algorithme est très fortement dépendant de ses paramètres initiaux, qui ne permettent de trouver

que des primitives à une échelle donnée. Il est alors nécessaire d'itérer le processus avec un autre paramètre d'échelle, pouvant parfois être complexe à choisir.

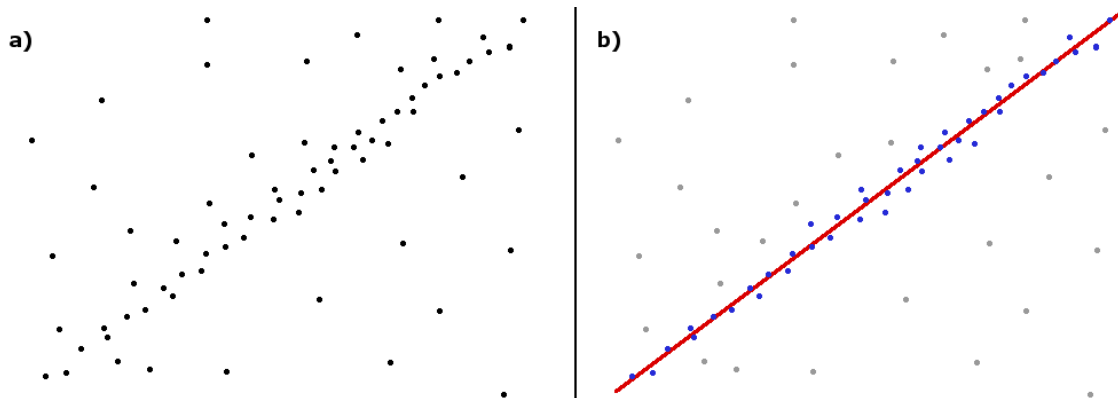


FIGURE 1.39 – Robustesse de l'algorithme RANSAC : a) Jeu de données avec valeurs aberrantes, b) Ligne ajustée. Les valeurs aberrantes n'ont aucune influence sur le résultat (points grisés).

Par la suite, plusieurs améliorations de cet algorithme ont été proposées afin de compenser ses lacunes. Ainsi, la méthode de Schnabel *et al.* [93] repose sur la probabilité d'avoir déjà trouvé la primitive optimale afin de réduire le nombre d'itérations et donc les temps de calcul. Ils proposent également d'utiliser un Octree pour guider le choix des points initiaux et ainsi accélérer encore le processus. Malheureusement, comme pour RANSAC, la méthode reste très dépendante de ses paramètres. Par ailleurs, Yang *et al.* [120] proposent de calculer le facteur d'échelle automatiquement afin d'être moins dépendant de paramètres experts. Cependant, leur méthode n'est pas capable de traiter des objets trop complexes, qui nécessitent d'être au préalable segmentés par l'utilisateur.

D'autres méthodes basées sur RANSAC vont également plus loin, comme celle de Li *et al.* [63]. Cette méthode utilise RANSAC pour extraire des primitives de manière hiérarchique, tout en les ajustant les unes par rapport aux autres. Cette dernière étape, permettant de prendre en compte des contraintes géométriques entre les primitives, est expliquée plus en détails dans la section 2.4.

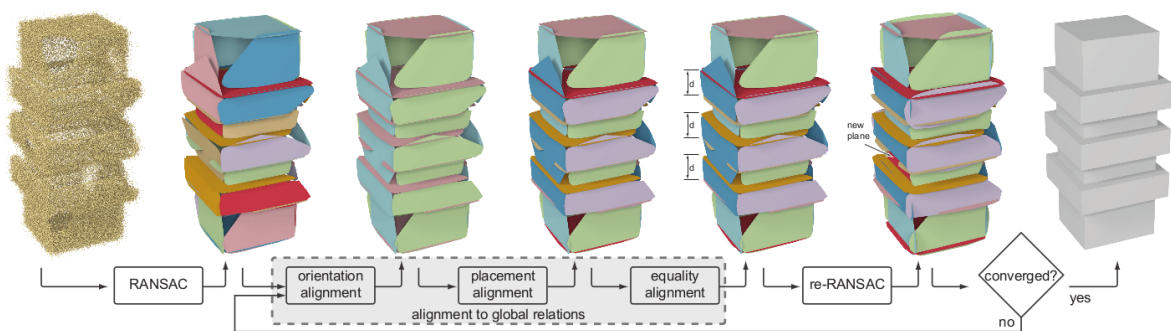


FIGURE 1.40 – Schéma global de la méthode de Li *et al.* [63] utilisant l'algorithme RANSAC.

Synthèse

Les méthodes directes sont souvent plus robustes aux imprécisions. Cependant, cette robustesse est souvent due à une approche itérative, engendrant pour la plupart des temps de calcul importants, ce qui n'est pas toujours adapté à une application industrielle. De plus, elles ne permettent pas toujours de retrouver les primitives associées à peu de points de manière fiable, et peuvent parfois échouer à identifier des primitives simples.

1.4 Synthèse

Il existe deux grandes familles d'approches permettant d'extraire les primitives géométriques contenues dans un maillage ou un nuage de points.

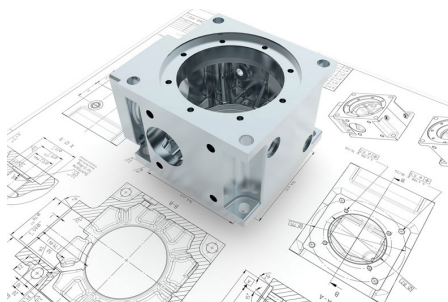
Les méthodes incrémentales commencent d'abord par appliquer une segmentation. Cette dernière permet de séparer les zones homogènes correspondant aux primitives et ainsi faciliter leur extraction. Ensuite, plusieurs techniques permettent de caractériser la forme globale ou locale de chaque sous-maillage, et il semble que le calcul des courbures soit le plus adapté. Enfin, les primitives peuvent être approximées globalement ou localement, à partir des sous-maillages et de la courbure.

Les méthodes directes tentent d'extraire les primitives directement ou globalement à partir du maillage, en procédant soit par sous-ensembles de points définis aléatoirement, soit par propagation à partir d'un point de départ. Celles-ci ont en général besoin de peu d'informations, et sont donc particulièrement adaptées aux nuages de points.

L'approche incrémentale permet souvent d'assurer une meilleure couverture du maillage par des primitives, ainsi qu'une meilleure identification de celles-ci. D'un autre côté, l'approche directe permet en général une meilleure robustesse aux imprécisions. Cependant, nous avons vu que cette dernière peut impliquer des temps de calcul plus longs et des problèmes d'identification. Il semble donc que l'approche incrémentale soit la plus adaptée à notre problématique.

Il semble qu'aucune méthode existante ne propose d'utiliser les avantages de ces deux approches, notamment en caractérisant la forme globalement et localement. De même, toutes les méthodes présentées ont des difficultés à gérer les maillages issus de numérisation.

La méthode d'extraction de primitives géométriques présentée au chapitre 3 tente de répondre à ces problématiques. Celle-ci, essentiellement incrémentale, est conçue pour être automatique, rapide, robuste et adaptative. L'objectif est de pouvoir traiter des maillages issus de numérisation avec des caractéristiques très différentes, sans nécessiter l'action d'un utilisateur expert.



Chapitre 2

Analyse et embellissement d'un ensemble de primitives pour la rétro-ingénierie

*« J'ai toujours rêvé que mon ordinateur soit aussi simple à utiliser que mon téléphone.
Ce rêve est devenu réalité : je ne comprends plus comment utiliser mon téléphone. »*

Bjarne Stroustrup

Résumé

Plusieurs objectifs sont envisageables dans le cadre de l'analyse d'un ensemble de primitives. Dans le cas de la rétro-ingénierie, nous cherchons à retrouver les informations induites par le processus de modélisation d'un objet. Ces informations peuvent par la suite permettre de mesurer des déviations ou encore guider une étape de régularisation du modèle.

Après une introduction sur la reconstruction de modèles à partir de primitives géométriques extraites d'un maillage 3D (section 2.1), nous présentons les deux types d'informations induites par la modélisation. Ainsi, les systèmes de référence en CAO sont détaillés dans la section 2.2. La section 2.3 décrit ensuite les trois types de motifs particuliers fréquemment présents dans les objets manufacturés. La section 2.3.1 détaille les approches permettant d'identifier les sous-ensembles de primitives caractéristiques formant des entités. La section 2.3.2 décrit plusieurs algorithmes permettant de reconnaître les répétitions de composants semblables au sein d'un objet. La section 2.3.3 présente les méthodes de détection des symétries. Enfin, le processus de régularisation des primitives grâce à ces informations, appelé embellissement, est présenté dans la section 2.4, suivi d'une synthèse (section 2.5).

Sommaire

2.1 Introduction	40
2.2 Reconnaissance d'un système de référence en CAO	43
2.3 Reconnaissance de motifs particuliers	46
2.3.1 Entités	47
2.3.2 Répétitions de géométrie	52
2.3.3 Symétries	53
2.4 Application à l'embellissement	56
2.5 Synthèse	59

2.1 Introduction

Nous avons vu dans le chapitre 1 comment extraire des primitives géométriques à partir d'un maillage 3D. Cependant, comme précisé dans l'introduction du manuscrit, l'objectif de la rétro-ingénierie est de reconstruire un modèle B-Rep complet. Ce dernier détaille alors la combinaison des primitives ainsi que leurs intersections permettant de délimiter les primitives par des contours. Or, les méthodes présentées dans le chapitre 1 permettent seulement d'extraire un ensemble de primitives indépendantes. La suite du processus consiste alors à définir des contours de l'objet, à partir d'un ensemble de primitives et du maillage segmenté utilisé pour ajuster ces dernières.

Une première classe de méthodes consiste à projeter les contours du sous-maillage ayant servi à ajuster chaque primitive sur celle-ci. Ensuite, ces contours discrets sont utilisés pour délimiter les primitives (Fig. 2.1). Cependant, chaque primitive est délimitée indépendamment des autres, ce qui ne permet pas de retrouver les informations structurelles présentes dans un modèle B-Rep. De plus, en particulier dans le cas d'un maillage issu de numérisation, les sous-maillages ne correspondent pas exactement aux primitives à cause des imprécisions. Par conséquent, les contours projetés peuvent être déconnectés, ce qui ne permet alors pas d'obtenir un modèle paramétrique topologiquement valide (Fig. 2.1.c). Dans le contexte de nos travaux, cette approche ne semble donc adaptée qu'à la visualisation.

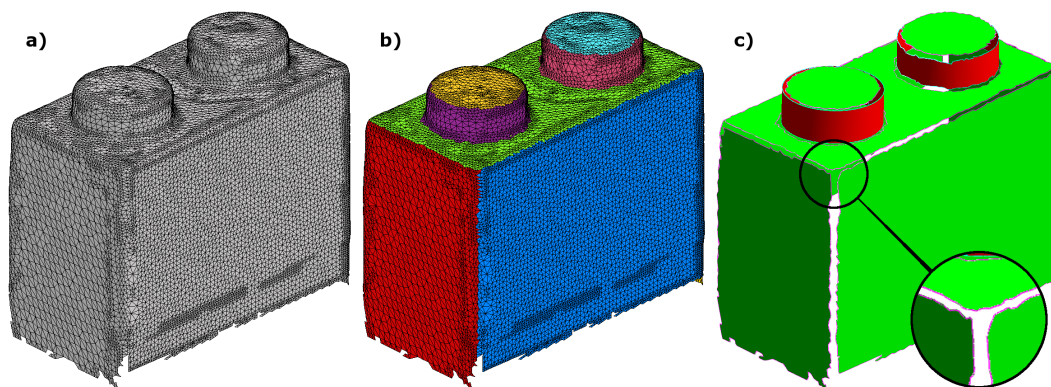


FIGURE 2.1 – Contours discrets : a) Maillage original, b) Sous-maillages ayant servi à ajuster les primitives, c) Délimitation par les contours des sous-maillages projetés sur les primitives.

Une seconde classe de méthodes consiste à utiliser des volumes et à construire un arbre CSG (Fig. 2.2), que nous avons décrit dans l'introduction du manuscrit. Cependant, nous ne connaissons pas *a priori* les volumes correspondant aux différentes combinaisons des primitives extraites à partir d'un maillage. De plus, les imprécisions du calcul des paramètres des primitives se répercutent sur les calculs d'union, de soustraction ou d'intersection réalisés, ce qui ne permet pas toujours de reconstruire un modèle topologiquement valide, par exemple dans le cas de tangences.

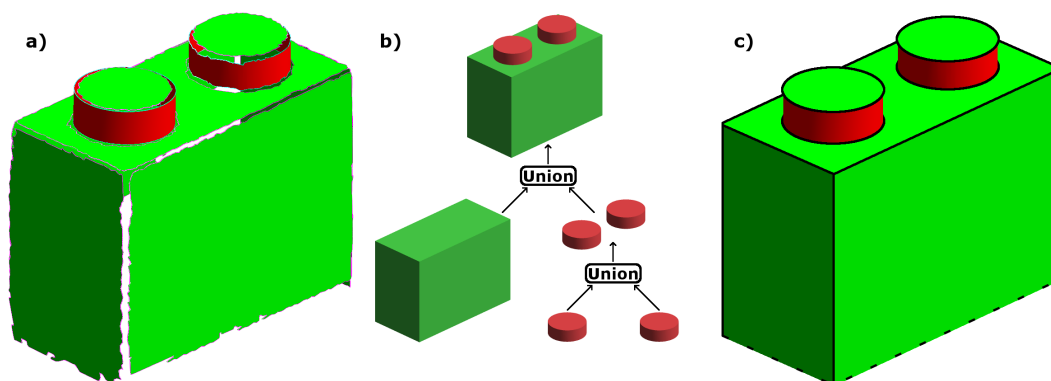


FIGURE 2.2 – Représentation volumique : a) Primitives, b) Arbre CSG, c) Primitives délimitées.

Fayolle et Pasko [33] proposent d'utiliser un algorithme génétique afin de reconstruire des arbres CSG correspondant aux données mesurées (Fig. 2.3). Ainsi, l'algorithme génétique calcule de nombreux arbres de construction du modèle contenant les primitives extraites. Ensuite, les arbres les plus proches du modèle sont sélectionnés et l'algorithme est itéré jusqu'à convergence. Enfin, l'arbre reconstruit correspond à une possibilité de construction de l'objet.

Cependant, les arbres CSG sont souvent très complexes même pour des objets relativement simples, comme illustré dans la figure 2.3.c. De même, les algorithmes génétiques impliquent souvent une complexité algorithmique importante. Dans le cadre de nos travaux, cette approche semble inadaptée au traitement d'objets complexes.

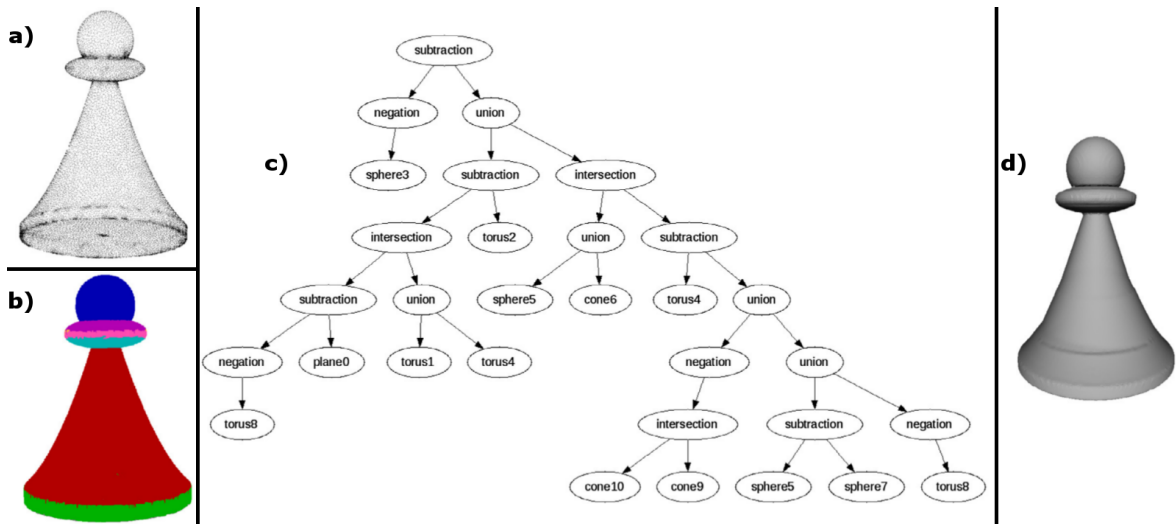


FIGURE 2.3 – Résultats de Fayolle et Pasko [33] : a) Points 3D, b) Primitives, c) Arbre CSG, d) Reconstruction. Pour un objet d'apparence peu complexe, l'arbre CSG est très compliqué.

La méthode proposée par Wang *et al.* [113] consiste à reconstruire un modèle à partir de volumes plus variés. Pour commencer, un volume global est construit grâce à l'enveloppe convexe de l'objet. Ensuite, les sous-volumes permettant de creuser la pièce sont définis par un utilisateur. Enfin, une succession d'opérations booléennes, essentiellement des soustractions avec le volume global, permettent de reconstruire pas à pas le modèle final (Fig. 2.4).

Cependant, cette approche implique de nombreuses opérations booléennes, parfois très complexes. De plus, le processus n'est pas automatique et nécessite un utilisateur expert.

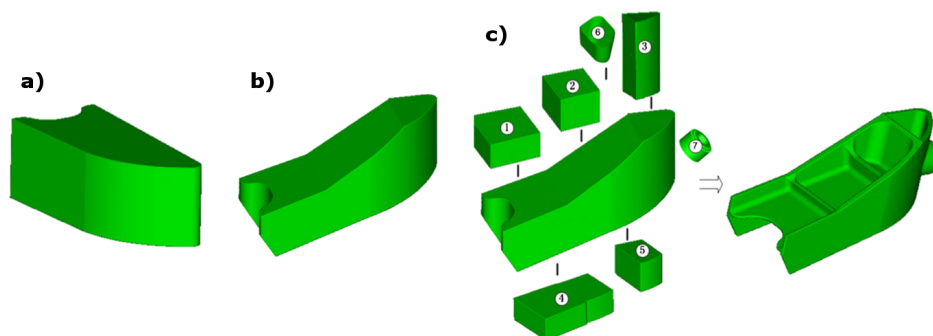


FIGURE 2.4 – Méthode de Wang *et al.* [113] : a) Bloc initial, b) Bloc intermédiaire, c) Reconstruction.

Une troisième classe de méthodes consiste à n'utiliser que les primitives et à calculer les intersections entre les primitives voisines (Fig. 2.5). Cependant, les imprécisions du calcul des paramètres des primitives se répercutent sur les intersections calculées. Par exemple, des primitives tangentes peuvent devenir disjointes ou des arêtes formant un coin peuvent ne plus s'intersecter (Fig. 2.5.c). Dans ce cas, le modèle n'est pas topologiquement valide et ne peut être reconstruit.

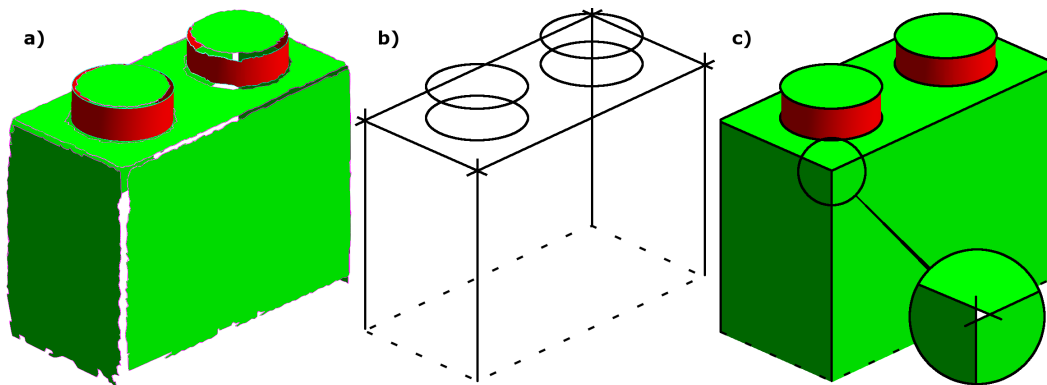


FIGURE 2.5 – Contours continus : a) Primitives, b) Intersections des primitives, c) Primitives délimitées.

En particulier, Bénéière *et al.* [7] développent une méthode basée sur le calcul des intersections entre primitives voisines (Fig. 2.6). Tout d'abord, un graphe d'adjacence des primitives est construit par propagation sur les points du maillage (Fig. 2.6.2). Ensuite, le graphe d'adjacence est analysé afin de déterminer les incohérences et délimiter les courbes d'intersection entre les primitives (Fig. 2.6.3). Pour cela, un ensemble de points d'intersection entre ces courbes est construit, où chaque point ne peut que correspondre à l'intersection de trois primitives, ni plus, ni moins. De plus, ces trois primitives doivent former un cycle dans le graphe, étant toutes voisines par ce même point. Les contours non délimités sont par la suite éliminés (Fig. 2.6.4), et les contours restants permettent de reconstruire le modèle B-Rep (Fig. 2.6.5).

Cependant, cette méthode ne permet pas de construire un modèle fermé s'il manque une primitive. En effet, une primitive manquante ne permet ni de détecter les cycles correspondant dans le graphe, ni de construire les points d'intersection délimitant les contours des primitives. Or, la numérisation d'une pièce peut être incomplète. De plus, les méthodes présentées dans le chapitre 1 ne garantissent pas que toutes les primitives soient extraites correctement à partir d'un maillage imprécis. Cette méthode nécessite donc une adaptation dans le cadre de nos travaux.

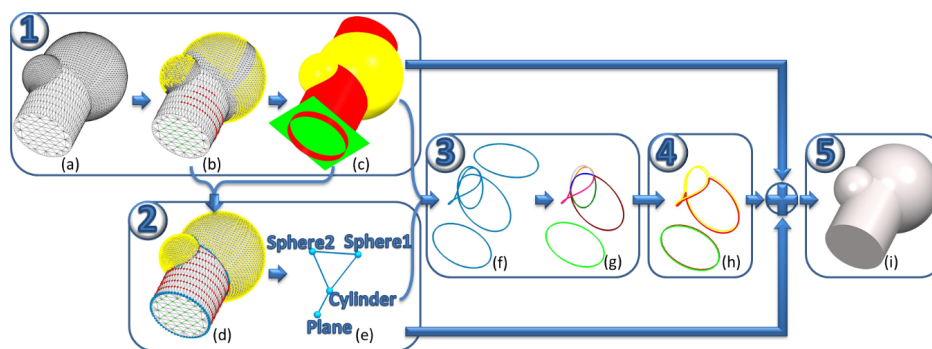


FIGURE 2.6 – Méthode de Bénéière *et al.* [7] : 1) Extraction de primitives, 2) Graphe d'adjacence, 3) Sélection des intersections, 4) Construction des contours, 5) Reconstruction du modèle final.

Synthèse

Dans la littérature, il existe trois classes de méthodes permettant de délimiter les primitives géométriques et ainsi reconstruire un modèle acceptable. La première utilise les contours des sous-maillages à partir desquels les primitives sont extraites. Cependant, ces derniers sont discrets et ne correspondent pas aux délimitations induites par la modélisation CAO. La seconde consiste à construire un arbre CSG représentant directement l'ensemble des primitives. Cependant, cette approche nécessite de connaître les sous-volumes de l'objet et semble trop complexe pour être utilisée dans le cadre de nos travaux. La troisième classe regroupe la plupart des méthodes de rétro-ingénierie, qui tentent de reconstruire un modèle B-Rep en délimitant les primitives par les intersections avec leurs voisines. Cette troisième approche semble privilégiée car elle réalise un bon compromis entre précision et complexité.

Cependant, ces différentes approches ne permettent pas de gérer efficacement les imprécisions du calcul des paramètres des primitives. Par conséquent, ces primitives ne s'intersectent pas correctement et les modèles reconstruits ne sont pas topologiquement valides. Il semble donc nécessaire d'utiliser des connaissances extérieures afin, par exemple, de régulariser les paramètres des primitives et ainsi calculer de meilleures intersections.

2.2 Reconnaissance d'un système de référence en CAO

Comme nous l'avons vu précédemment, la reconstruction d'un modèle B-Rep peut être confrontée à des erreurs dues aux imprécisions. Nous cherchons alors à extraire des informations extérieures à partir de l'ensemble des primitives, afin d'améliorer cette reconstruction. Or, en CAO, la modélisation d'une pièce s'appuie généralement sur un système de référence comprenant :

- un repère tridimensionnel de référence ;
- des plans, axes et points de référence ;
- des règles de dimensionnement et tolérancement géométriques (GD&T) [66].

Ces éléments de référence permettent respectivement de définir l'orientation, l'emplacement et les dimensions d'une pièce et de ses constituants (Fig. 2.7). Les GD&T contiennent également un ensemble de tolérances géométriques et topologiques, liant la conception à la fabrication. Ce système facilite ainsi la conception d'une pièce dont les primitives respectent certaines relations géométriques telles que le parallélisme, l'orthogonalité, la concentricité ou la tangence. De manière plus locale, certaines primitives peuvent parfois être utilisées comme référence pour contraindre les suivantes, par exemple pour une extrusion à partir d'un plan.

Dans le cadre de la rétro-ingénierie, il semble intéressant de retrouver un tel système de référence à partir des primitives extraites et du maillage sous-jacent.

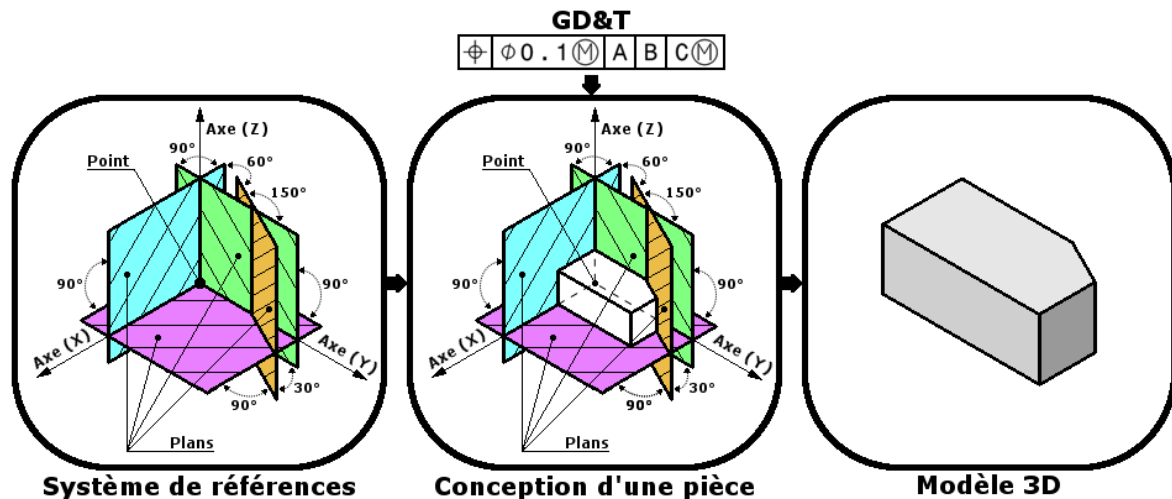


FIGURE 2.7 – Exemple de conception CAO basée sur un système de référence.

Une première approche consiste à sélectionner une primitive de référence, à partir de laquelle il est possible d'extraire une orientation et une position de référence. Par exemple, Campana et Germani [19] proposent une méthode permettant d'extraire des plans et axes de référence à partir d'un nuage de points. Celle-ci s'inscrit dans un processus d'inspection de pièces après fabrication. Ainsi, une extraction de primitives est réalisée et chaque primitive est considérée successivement comme la primitive de référence. Cette dernière est alors utilisée pour réaliser des mesures de déviation avec les autres surfaces, afin de quantifier les erreurs de fabrication.

Cependant, l'utilisation d'une primitive unique comme élément de référence ne permet pas de mesurer ces déviations de manière fiable. Comme nous pouvons le voir dans la figure 2.8, les

erreurs mesurées sont différentes en choisissant le plan horizontal ou le plan vertical. Dans cet exemple, il apparaît clairement que la déviation cumulée est beaucoup plus importantes dans le cas B que dans le cas A. Logiquement, dans ce cas, le référentiel qui prévaut devrait correspondre au repère orthonormé de référence, malheureusement inconnu *a priori*. Enfin, cette inspection n'est réalisée que localement, alors que la modélisation peut produire des relations géométriques globales.

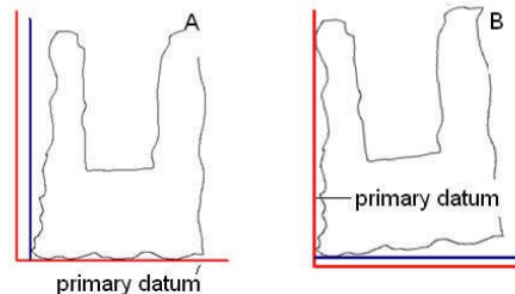


FIGURE 2.8 – Problème du choix de la primitive de référence [19] : a) Plan horizontal, b) Plan vertical.

Une seconde approche consiste à extraire des informations extérieures telles qu'un repère orthonormé, à partir d'un sous-ensemble de primitives. Ainsi, Wu et Gu [119] proposent de construire un repère de référence à partir d'un sous-ensemble de trois primitives sélectionnées par l'utilisateur. Chaque primitive de référence est alors décomposée en éléments de bases, à savoir des plans, des axes et des points. Ensuite, le repère de référence est reconstruit de manière à respecter les relations d'orthogonalité et d'intersection entre ces éléments. Prenons l'exemple de deux cylindres et un plan tous parallèles deux à deux. Leur orientation commune définit une première orientation de référence. La seconde orientation de référence est déterminée par l'alignement des deux cylindres, puis le troisième est calculée par le produit vectoriel des deux premières. Enfin, l'origine du repère correspond à l'intersection du plan sélectionné et de l'axe du premier cylindre sélectionné.

Cependant, la méthode se base sur des modèles sans imprécisions. De plus, elle dépend fortement d'un utilisateur pour définir le sous-ensemble de primitives dont l'ordre de sélection fournit des résultats différents : l'axe principal du repère est par exemple défini par l'orientation de la première primitive sélectionnée. Cette méthode n'est donc pas automatique et ne permet pas toujours de construire un repère lorsque l'utilisateur ne choisit pas les bonnes primitives, comme c'est le cas pour trois sphères ne fournissant aucune orientation par exemple.

Une troisième approche consiste à analyser l'ensemble des primitives afin d'estimer des références extérieures de manière globale. A partir d'un modèle sans imprécisions, Mohan *et al.* [75] proposent de reconnaître automatiquement des GD&T. La première phase consiste à détecter les différentes orientations de contrôle à partir des axes et normales des primitives. Ainsi, les orientations parallèles sont regroupées afin de calculer les orientations principales. La seconde phase consiste à arranger les primitives le long de ces axes de contrôle afin d'obtenir des positions relatives. Cela permet alors de retrouver et d'afficher les dimensionnements des pièces.

Cependant, cette méthode se base sur des modèles contenant peu d'imprécisions. De plus, l'analyse ne porte que sur le parallélisme alors que d'autres relations géométriques existent entre les primitives. Par exemple, la concentricité n'est pas abordée. Notons tout de même l'importance de se baser sur des ensembles de primitives plutôt qu'une seule, permettant d'éviter la propagation d'erreur.

A partir de maillages issus de numérisation, Le et Duan [59] proposent d'estimer les orientations principales des primitives. Pour cela, ils construisent une sphère Gaussienne à partir des normales des points et recherchent des *pôles*, correspondant à des zones très denses de la sphère.

Ensuite, ils construisent une seconde sphère en choisissant aléatoirement des triplets de la première. Chaque triplet représente alors un triangle dont la normale est ajoutée à la seconde sphère. Les pôles de cette dernière sont alors extraits, puis associés à des cercles sur la première sphère. Ces cercles correspondent ainsi à des structures cylindriques dans le maillage. Celles-ci définissent alors des orientations de contrôle, représentant des axes de référence (Fig. 2.9).

Cependant, cette méthode fonctionne uniquement sur des objets contenant des plans et des cylindres. En effet, les autres types de primitive ne sont pas pris en compte dans l'étude. Nous pouvons également nous questionner sur l'efficacité de cette approche lorsque l'objet contient des primitives comme des sphères ou des tores, dont les normales se distribuent uniformément sur toute la sphère Gaussienne.

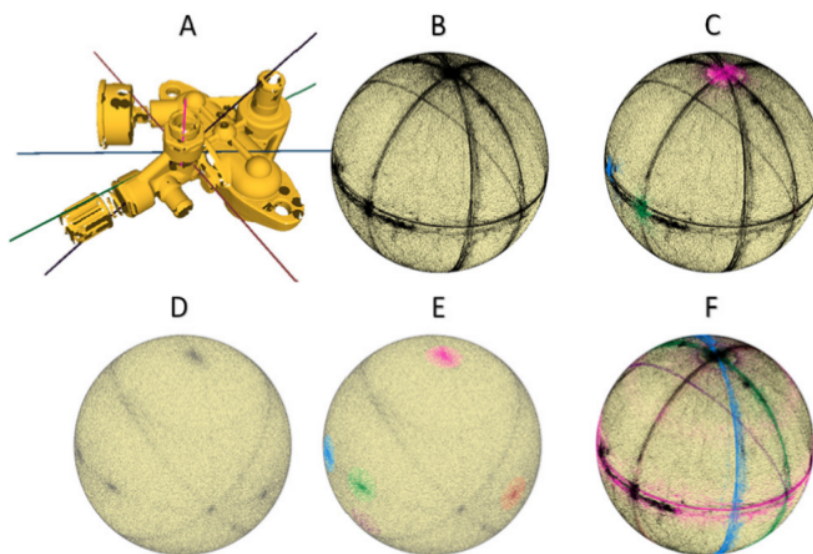


FIGURE 2.9 – Résultats de Le et Duan [59] : a) Modèle initial avec les orientations de référence, b) Sphère Gaussienne du maillage, c) Zones denses détectées, d) Sphère Gaussienne échantillonnée, e) Zones denses détectées sur la seconde sphère, f) Cercles extraits indiquant les orientations principales (cylindres).

Kovács *et al.* [54] proposent de construire des repères tridimensionnels locaux à partir d'un ensemble de primitives extraites d'un maillage issu de numérisation. Pour cela, une sphère Gaussienne est construite à partir des orientations des plans extraits. Ensuite, les orientations sont regroupées lorsqu'elles sont détectées comme parallèles, selon une tolérance. Enfin, chaque triplet d'orientations orthogonales deux à deux définit un repère tridimensionnel. Ainsi, comme illustré dans la figure 2.10, un objet peut être construit selon plusieurs repères possibles.

Cependant, les orientations principales ne sont détectées qu'à partir des plans alors que les objets utilisés contiennent d'autres primitives. De plus, de nombreux repères peuvent être construits sans pour autant définir une priorité sur ces derniers. Il serait alors intéressant de reconnaître un repère unique, comme c'est le cas en modélisation CAO.

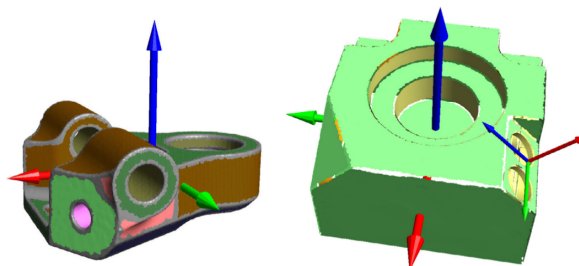


FIGURE 2.10 – Résultats de Kovács *et al.* [54] : plusieurs repères de référence indépendants sont extraits.

L'article de Weißgerber *et al.* [116] met également en avant le besoin d'utiliser des systèmes de référence communs pour la conception d'assemblages. Ils prennent pour exemple une jonction de moteur à combustion, mettant en jeu un joint déformable (Fig. 2.11). Ils montrent que les tolérances pour chaque élément de l'assemblage doivent être définies de sorte que les deux joints fassent bien contact afin d'éviter toute fuite. Les deux systèmes de référence doivent alors être définis de manière cohérente. Il semble donc nécessaire, dans le cadre de nos travaux, de définir un repère tridimensionnel global afin de s'assurer de la cohérence de toutes les primitives.

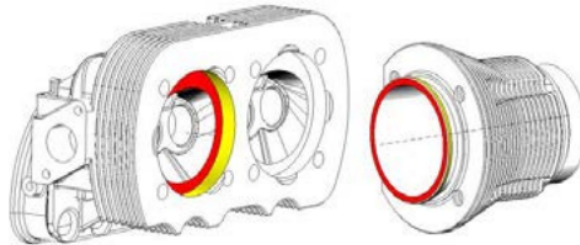


FIGURE 2.11 – Exemple de Weißgerber *et al.* [116] montrant le besoin de systèmes de référence cohérents. Des primitives critiques sont ici prises comme références afin de contraindre l'assemblage.

Synthèse

Plusieurs approches proposent d'extraire des éléments de référence à partir d'un ensemble de primitives ou d'un maillage. La plupart d'entre elles recherchent les orientations principales de l'objet, mais peu proposent d'extraire des informations positionnelles, comme par exemple des plans de référence. La définition d'un système de référence semble alors commencer par la reconnaissance d'un repère orthonormé contenant les orientations principales de la pièce. Cependant, ces méthodes sont soit basées sur des modèles sans imprécisions, soit ne prennent pas en compte toutes les primitives de l'objet. En effet, nous avons vu que se baser sur un sous-ensemble restreint de primitives ne permet pas de calculer un repère de référence de manière fiable.

Il semble alors nécessaire de concevoir une méthode permettant d'extraire un repère de référence de manière globale, à partir de l'ensemble des primitives extraites. Ensuite, il serait intéressant de retrouver des plans, axes et points de référence, également de manière globale afin d'éviter une éventuelle propagation d'erreur. Enfin, des règles de dimensionnement pourraient définir des relations supplémentaires entre les primitives, telles que des rayons égaux par exemple.

2.3 Reconnaissance de motifs particuliers

Lors de la modélisation d'une pièce manufacturée, il est fréquent de construire des structures géométriques particulières afin, par exemple, de donner une fonction mécanique précise à la pièce. Ces structures sont communément appelées *motifs*, et correspondent à une combinaison de primitives respectant des relations géométriques. Dans le cadre de la rétro-ingénierie, reconnaître ces motifs particuliers permettrait d'en déduire les relations entre les primitives et de retrouver des informations de conception et de fabrication de l'objet. Dans cette section, nous présentons trois grandes classes de motifs.

La première classe regroupe les sous-ensembles de primitives voisines formant une structure géométrique liée à un processus de fabrication ou un outil spécifique [95, 91]. Ces dernières correspondent alors à des *entités* [29, 47], pouvant être des perçages particuliers par exemple. Des méthodes de reconnaissance des entités sont présentées dans la section 2.3.1.

La seconde classe regroupe les répétitions d'une structure géométrique de référence, pouvant contenir une ou plusieurs primitives [108]. Cette *répétition de géométrie* peut alors se construire selon un schéma prédéfini, par exemple une droite ou un cercle. La particularité principale de ces motifs est l'équidistance de deux structures géométriques successives. Ainsi, une répétition

circulaire positionne les exemplaires de la géométrie de référence de manière uniforme sur 360°. Des méthodes de reconnaissance des répétitions sont décrites dans la section 2.3.2.

La troisième classe regroupe les différentes *symétries* [73], pouvant être planaires, axiales ou centrales. Ces dernières peuvent être globales ou locales en fonction des primitives mises en jeu. Des méthodes de détection des symétries sont ainsi détaillées dans la section 2.3.3.

2.3.1 Entités

Han *et al.* [46] et Babic *et al.* [3] présentent deux états de l'art sur la reconnaissance d'entités à partir de modèles B-Rep. Il semble alors que les trois approches les plus fréquemment utilisées soient basées sur des graphes, des décompositions volumiques, ou d'autres informations.

La première approche consiste à construire un graphe de relations entre les primitives, fondé sur l'adjacence [82, 103] et des relations géométriques telles que le parallélisme ou la concentricité [118]. Ensuite, la reconnaissance consiste à détecter, au sein du graphe de la pièce, des sous-graphes caractéristiques correspondant à la géométrie particulière des entités.

La seconde approche consiste à décomposer la pièce en sous-volumes indépendants, à partir des informations contenues dans le modèle B-Rep. Ensuite, la reconnaissance consiste à détecter des sous-ensembles de volumes correspondant à des entités.

La troisième approche regroupe les autres méthodes, pouvant par exemple se baser sur des informations topologiques ou des sections 2D. Ces approches sont moins généralistes et permettent de traiter des cas plus complexes, en s'aidant de toutes les informations disponibles.

Enfin, les méthodes récentes sont souvent dites *hybrides*, car elles combinent plusieurs de ces approches. D'après les auteurs, il semblerait que toutes ces méthodes aient tout de même une complexité relativement élevée. D'autres approches sont également possibles, telles que l'utilisation de réseaux de neurones [52, 81] ou encore des algorithmes génétiques [83]. Cependant, ces dernières nécessitent une base importante de données annotées que nous n'avons pas dans le cadre de nos travaux. Elles ne seront donc pas détaillées.

Méthodes basées sur la construction d'un graphe

Les méthodes qui se basent sur un graphe d'adjacence ou de relations géométriques ne permettent généralement pas de détecter correctement les entités en intersection. En effet, les intersections modifient les sous-graphes associés, par exemple en créant des adjacences supplémentaires ou en supprimant des primitives lors d'un enlèvement de matière. Pour résoudre ce problème, Rameshbabu et Shunmugam [87] proposent une méthode utilisant le nombre de voisins de chaque primitive pour simplifier les graphes construits. Ainsi, nous pouvons voir qu'à partir du modèle de la figure 2.12.a et du graphe associé (Fig. 2.12.b), il est possible de construire des primitives, qui ne sont pas dans le modèle initial, représentant les cavités de l'objet (Fig. 2.12.c). Ainsi, le graphe de ces cavités étant beaucoup plus simple (Fig. 2.12.d), il permet de reconnaître des entités plus facilement, voir même d'en identifier une variété plus importante.

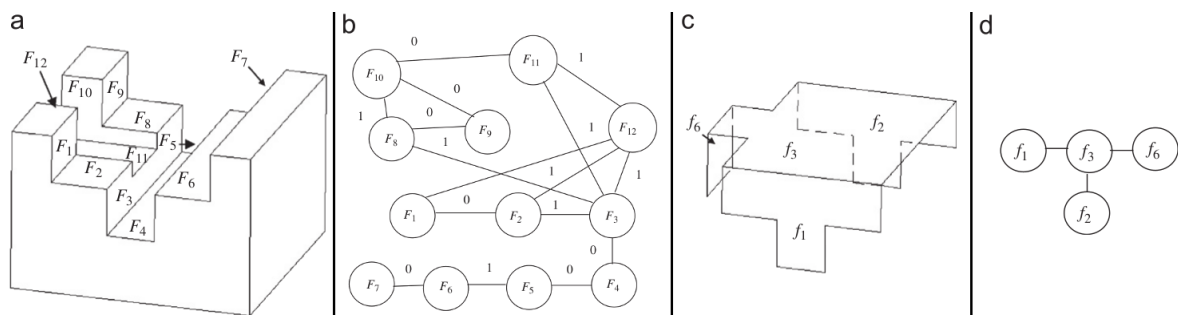


FIGURE 2.12 – Approche de Rameshbabu et Shunmugam [87] : a) Modèle initial, b) Graphe construit avec la méthode de Joshi et Chang [51], c) Cavité extraite, d) Graphe de la cavité extraite.

Sunil *et al.* [102] proposent une autre méthode permettant de reconnaître les structures correspondant à des enlèvements de matière, telles que les poches ou les rainures. Pour cela, ils commencent par extraire des sous-graphes, à partir du graphe d'adjacence, en se basant sur le constat qu'un enlèvement de matière est souvent défini par un plan formant la base, relié à toutes les autres primitives par des arêtes concaves. Ainsi, l'opération consiste principalement à retirer du graphe initial les relations d'adjacence correspondant à des arêtes convexes. Dans chaque sous-graphe obtenu, le plan formant la base est alors automatiquement déterminé comme le seul noeud voisin de tous les autres. Enfin, le nombre d'adjacences de chaque primitive permet de reconnaître différentes entités, en particulier en différenciant les poches et les rainures, débouchantes ou non (Fig. 2.13).

Cependant, les méthodes utilisant uniquement le graphe d'adjacence ne prennent pas en compte les relations géométriques entre les primitives, ce qui peut parfois poser un problème d'identification. En effet, la modélisation d'une rainure implique que la base soit orthogonale à tous les autres côtés, contrairement à une liaison glissière plus générale, en forme de trapèze par exemple, qui serait pourtant considérée comme une rainure.

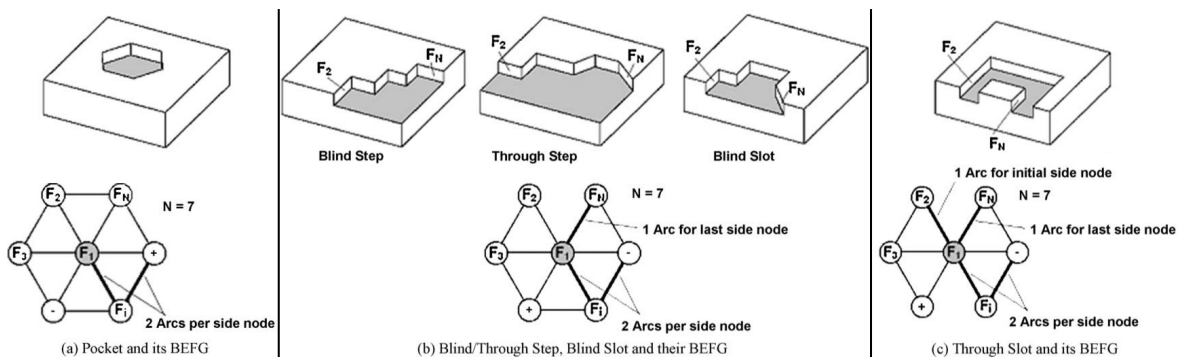


FIGURE 2.13 – Approche de Sunil *et al.* [102] permettant de différencier les poches ou rainures, débouchantes ou non, grâce à l'analyse des relations d'adjacence entre les primitives.

Lupinetti *et al.* [65] proposent également une méthode basée sur les graphes pour reconnaître les différentes parties des roulements à billes. La difficulté du problème réside principalement dans les variations entre les roulements à billes et leurs représentations en CAO (Fig. 2.14.a). En effet, le nombre et le diamètre des billes peut varier, de même que la forme de l'assemblage (avec ou sans rainure par exemple). La méthode est donc conçue pour définir plusieurs niveaux d'abstraction, dans lesquels l'ensemble des billes peut être représenté par un unique noeud dans le graphe, associé tantôt à un anneau, tantôt à des sphères séparées (Fig. 2.14.b). Dans les deux cas, toutes les billes sont alors considérées comme faisant partie d'un seul constituant de l'assemblage. Les auteurs montrent ainsi qu'il peut être difficile de représenter toute la variabilité des entités à travers des graphes d'adjacence uniquement.

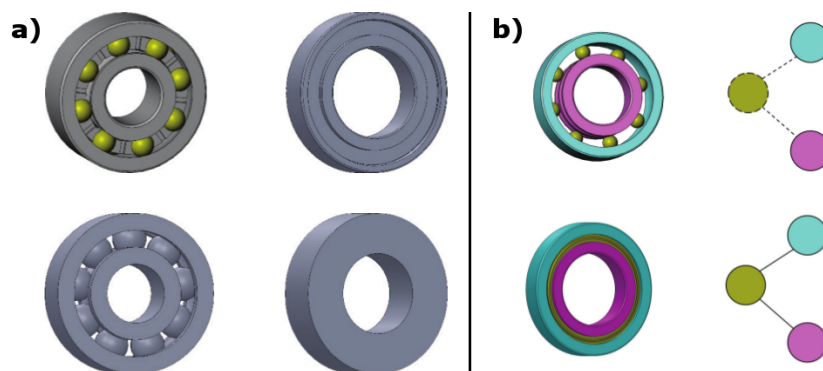


FIGURE 2.14 – Approche multi-résolution de Lupinetti *et al.* [65] : a) Représentation d'un roulement à billes à plusieurs résolutions, b) Graphe d'adjacence commun à toutes les représentations.

Méthodes basées sur une décomposition volumique

La seconde approche consiste à appliquer une décomposition volumique de l'objet, à partir de cellules unitaires ou d'enveloppes convexes. Cette approche semble historiquement liée aux arbres CSG présentés dans l'introduction du manuscrit. Les méthodes basées sur des cellules unitaires commencent par extraire un volume correspondant à un enlèvement de matière, grâce à la soustraction entre l'enveloppe convexe de l'objet et l'objet lui-même. Ensuite, ce volume est décomposé en un ensemble de cellules de même forme. Enfin, toutes les cellules adjacentes sont fusionnées afin de retrouver des volumes disjoints, qui peuvent être analysés séparément afin de reconnaître des entités.

Cependant, la forme et les dimensions des cellules doivent être paramétrées au préalable. La variabilité de ces cellules entraîne une multitude de compositions possibles permettant de reconstruire les volumes finaux (Fig. 2.15) : c'est *l'effet global de la géométrie locale* [46]. Cela pose alors des problèmes de complexité, de précision, mais aussi d'identification. En effet, les entités en intersection sont reconstruites en un seul et même volume.

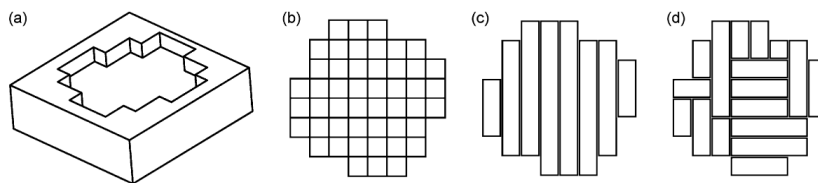


FIGURE 2.15 – Exemples de décompositions par cellules unitaires [3].

Geng *et al.* [43] présentent une méthode basée sur les enveloppes convexes, permettant de reconnaître des entités en interaction ou non dans un modèle B-Rep (Fig. 2.16). La première étape consiste à rechercher des poches, chacune définie par un plan et ses primitives voisines, qui doivent elles-mêmes être voisines entre elles deux par deux. Ce plan correspond alors à la base d'un volume correspondant à un enlèvement de matière. Or, ces volumes extraits ne sont pas toujours convexes, comme illustré dans la figure 2.16. Dans ce cas, les primitives voisines au plan formant la base sont utilisées pour décomposer le volume en sous-volumes convexes. Cette décomposition permet alors, en cas d'interaction entre plusieurs entités, de reconstruire les primitives qui n'existent plus dans le modèle. Enfin, les sous-volumes obtenus permettent de reconnaître des entités indépendantes, et de proposer un arbre de construction de la pièce.

Cependant, l'approche par décomposition volumique est souvent coûteuse en temps de calcul, en particulier à cause de la complexité non-linéaire des volumes mis en jeu. De plus, les pièces complexes contenant de nombreuses entités en interaction ne peuvent pas toujours être correctement reconstruites à cause des ambiguïtés engendrées par les volumes obtenus. Par exemple, la combinaison de deux volumes convexes peut ne pas engendrer un nouveau volume convexe.

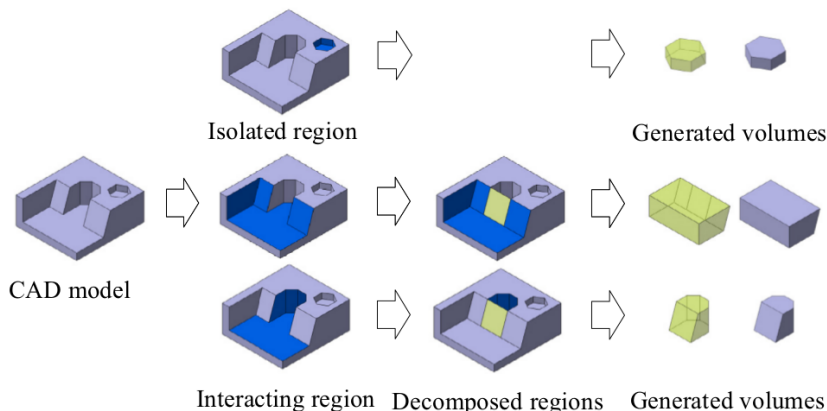


FIGURE 2.16 – Schéma global de la méthode de Geng *et al.* [43].

Méthodes basées sur des sections 2D

La méthode de Muraleedharan *et al.* [78] consiste à utiliser des sections 2D d'un modèle B-Rep pour reconnaître des profils caractéristiques des entités. L'algorithme commence par générer de nombreux plans de coupe, avec des positions et orientations différentes. Ensuite, les intersections entre l'objet et les plans sont calculées, ce qui permet de reconstruire des contours paramétriques 2D. Ces contours sont alors analysés afin de reconnaître des contours caractéristiques de la géométrie des entités (Fig. 2.17). Cette méthode permet d'identifier de nombreuses entités en même temps, puisque plusieurs contours peuvent être présents sur un seul et même plan.

Cependant, un grand nombre de plans est nécessaire afin d'obtenir des résultats satisfaisants. Ce nombre, paramétré à l'avance, dépend de la complexité et de l'échelle de l'objet, ce qui semble difficile à fixer pour un utilisateur non-expert. De plus, les intersections calculées et les analyses de contours se multiplient avec le nombre de plans, ce qui augmente encore la complexité de la méthode. Cette dernière semble alors difficilement utilisable dans le cadre de nos travaux.

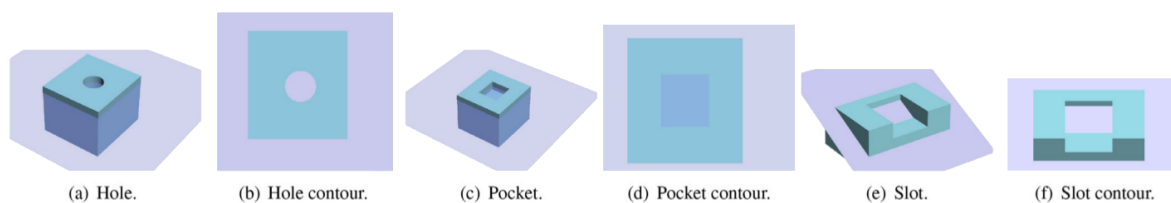


FIGURE 2.17 – Exemples de plans de coupe caractéristiques de Muraleedharan *et al.* [78].

Méthodes utilisant des systèmes de règles

Une quatrième classe de méthodes consiste à utiliser un système de règles afin de concevoir des algorithmes hiérarchiques et facilement extensibles. Ces règles peuvent alors être définies librement, en fonction d'un graphe, d'une décomposition volumique ou d'autres caractéristiques ou relations entre les primitives. En particulier, ces méthodes sont souvent couplées à un système d'ontologie, correspondant intuitivement à une grammaire, ou plus précisément à une manière de raisonner à partir d'un ensemble structuré de termes ou de concepts. Dans le cas de la rétro-ingénierie, ces règles peuvent fournir une hiérarchie de construction entre les entités. Par exemple, une rainure en forme de croix peut être représentées comme l'association de deux rainures linéaires orthogonales. Cette croix est donc une entité fille de l'entité rainure.

Wang et Yu [114] proposent par exemple de convertir un modèle B-Rep en un ensemble de primitives et une liste de règles permettant de reconstruire l'objet. Dans un premier temps, les primitives sont séparées en sous-ensembles en retirant les arêtes convexes, comme dans la méthode de Sunil *et al.* [102], formant alors des CCFS (Concavely Connected Face Set). Chaque CCFS est ensuite analysé via des règles prédéfinies afin de rechercher l'entité correspondante parmi toutes celles définies par une classification sous forme d'arbre d'héritage (Fig. 2.18).

Basée sur la même idée, la méthode de Niu *et al.* [79] consiste à construire et utiliser une base de données à partir de requêtes SQL. Nous avons également vu précédemment que Vilmart *et al.* [110] proposaient de déterminer des fonctions de composants en utilisant une base de connaissances permettant de faire des inférences en fonction des plans de symétrie extraits et des interfaces entre les composants d'un assemblage (Fig. 2.19).

Cependant, ces approches sont toutes conçues pour traiter des modèles B-Rep, qui contiennent des informations structurées et pas seulement un ensemble de primitives indépendantes, comme c'est le cas dans le cadre de nos travaux. De même, le calcul imprécis des paramètres des primitives implique un système de tolérances, parfois complexe à mettre en place avec ces méthodes. Enfin, des ambiguïtés de construction peuvent apparaître lorsque plusieurs procédés peuvent permettre d'aboutir au même objet, impliquant plusieurs arbres de construction possibles.

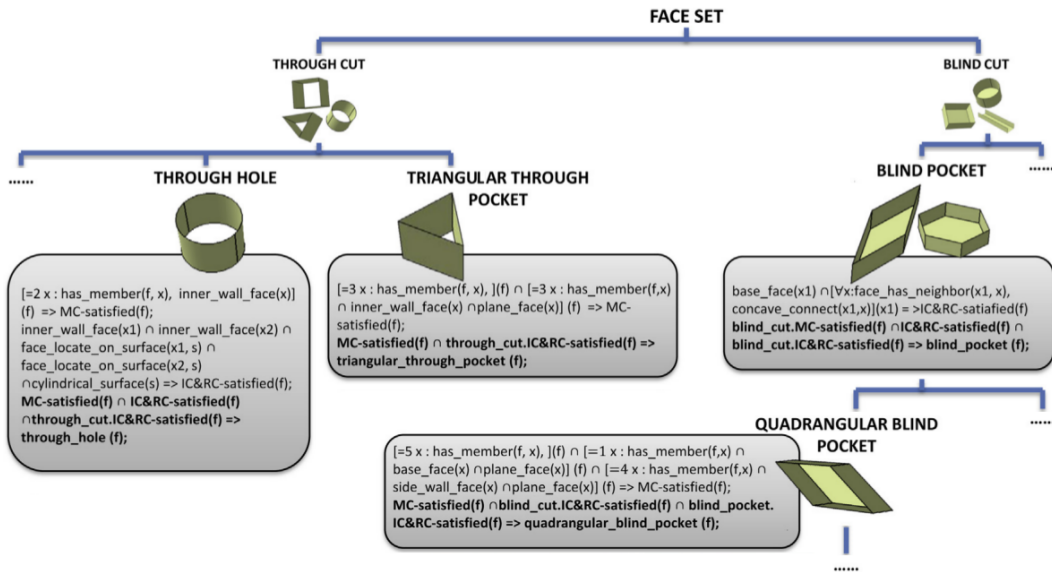


FIGURE 2.18 – Système de règles de Wang et Yu [114] sous forme d'arbre d'héritage.

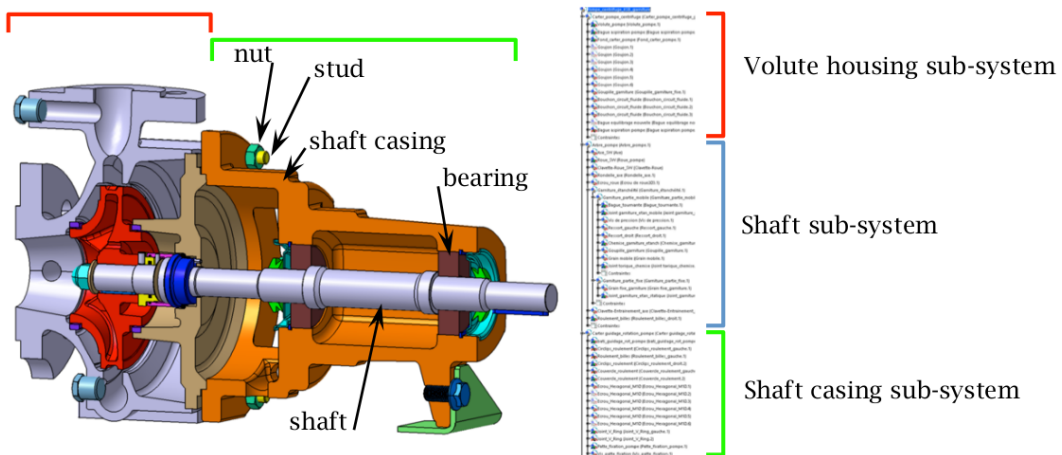


FIGURE 2.19 – Analyse fonctionnelle d'un modèle de pompe par Vilmart *et al.* [110].

Synthèse

Plusieurs approches permettent d'extraire des sous-ensembles de primitives correspondant à des entités. Trois d'entre elles semblent être les plus fréquemment utilisées. La première, basée sur des graphes, ne permet pas toujours de retrouver des entités complexes ou en intersection. Quelques méthodes hybrides proposent de compenser ce problème mais sont souvent spécifiques à certains types d'entité. La seconde, basée sur une décomposition volumique, souffre souvent d'une complexité plus élevée et d'un problème de séparation de certains volumes en intersection représentant plusieurs entités, que nous ne connaissons pas a priori. La troisième, basée sur d'autres informations calculées à partir de l'objet telles que des sections 2D, semble plus rapide mais aussi moins généraliste. En effet, certaines informations peuvent être spécifiques à un type d'entité. En parallèle de ces trois approches, plusieurs méthodes sont basées sur des systèmes de règles. Ces dernières, qui sont aussi les plus récentes, semblent obtenir les meilleures performances. De plus, la hiérarchie et la diversité des règles mises en place en font une approche privilégiée par son extensibilité et sa généralisation. Dans la littérature, il semble que très peu de méthodes recherchent des entités à partir de primitives extraites d'un maillage 3D numérisé. Il serait intéressant d'adapter une méthode basée sur un système de règles, afin de prendre en compte un maximum d'information déduite du maillage et des primitives extraites, telles que les relations géométriques par exemple.

2.3.2 Répétitions de géométrie

Pauly *et al.* [85] présentent un état de l'art sur la détection de motifs réguliers dans les maillages 3D et les modèles B-Rep. Ces motifs peuvent être générés lors de la modélisation CAO, par répétition d'un sous-ensemble de primitives déjà construit. Ces répétitions sont principalement basées sur des translations, des rotations, ou une combinaison des deux (Fig. 2.20). Des motifs plus complexes peuvent aussi mettre en jeu des changements d'échelle, mais sont généralement très peu utilisés en modélisation. Nous pouvons également remarquer que ces motifs, correspondant à des relations entre les primitives, sont définis à partir de paramètres tels qu'un axe et des distances, éventuellement reliés au système de référence (voir section 2.2).

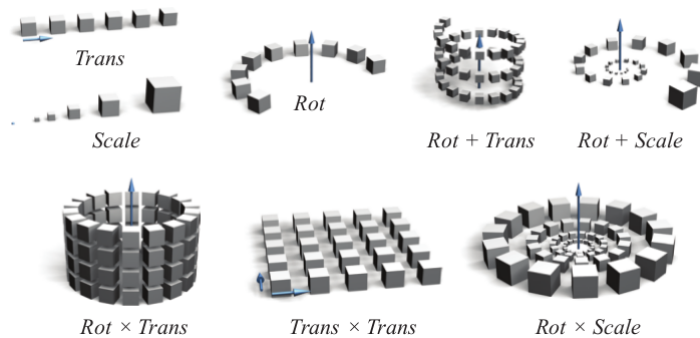


FIGURE 2.20 – Répétitions présentées par Pauly *et al.* [85] comme des combinaisons de translations, rotations et changements d'échelle. Ces dernières sont définies par un axe et des distances.

Vilmart *et al.* [110] proposent de détecter des répétitions linéaires et circulaires de sous-ensembles identiques de primitives (Fig. 2.21). Ces répétitions sont automatiquement déduites des plans de symétrie extraits au préalable (voir section 2.3.3). En effet, certaines combinaisons caractéristiques de plans de symétrie peuvent permettre d'en déduire des répétitions. Par exemple, une symétrie définie par des plans partageant un axe commun correspond à un motif circulaire (Fig. 2.21.a). De même, un ensemble de plans de symétrie parallèles et orthogonaux permet par exemple de retrouver des répétitions linéaires (Fig. 2.21.b).

Cependant, cette méthode est limitée à deux types de répétitions. Cette dernière, conçue pour traiter des modèles B-Rep, n'est pas utilisable directement dans le cadre de nos travaux.

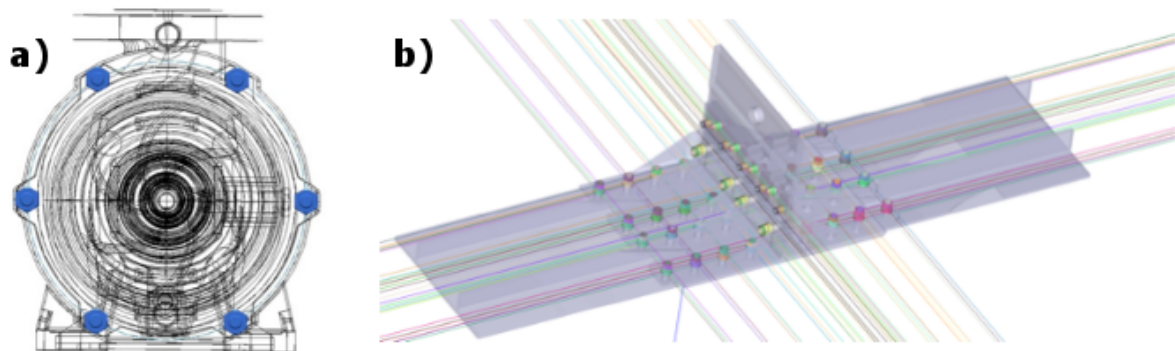


FIGURE 2.21 – Répétitions extraites par Vilmart *et al.* [110] : a) Circulaires, b) Linéaires.

Urbanic et Elmaraghy [108] proposent de se baser sur des sections 2D d'un nuage de points issu de numérisation. Pour commencer, l'algorithme construit un ensemble de coupes de l'objet selon trois directions orthogonales, correspondant à un repère de référence. Ensuite, les points 3D sont projetés sur les coupes les plus proches, puis un ensemble de contours 2D est reconstruit. A partir de ces contours, les auteurs recherchent une variété très large de motifs (Fig. 2.22). Cette recherche est ordonnée et tente de regrouper des contours similaires. La méthode commence par

rechercher les sous-ensembles de contours dont les centres sont alignés sur une droite ou inscrits dans un cercle. Les droites et arcs de cercle obtenus définissent alors des répétitions linéaires et circulaires. Les centres des répétitions et des contours isolés sont par la suite placés dans une grille, afin de rechercher des combinaisons de plusieurs motifs. Le processus est alors itéré jusqu'à ne plus détecter de nouveau motif.

Cependant, l'ordre prédéfini pour la recherche ne permet pas de détecter toutes les combinaisons de motifs. En effet, dès qu'un motif est trouvé, ses éléments constitutants ne sont plus pris en compte dans les prochaines analyses. Par exemple, une répétition circulaire de huit éléments peut être représentée par deux répétitions rectangulaires de quatre éléments. Dans ce cas, les motifs extraits dépendront de l'ordre de recherche et non de l'intention de conception.


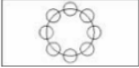



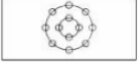
Pattern	Pictorial	Parameters
Linear		$N, \Delta x, \Delta y, \theta, \text{start pt } (x_s, y_s)$
Circular		$N, R, \Delta \phi, \theta, \text{start pt } (x_s, y_s), \text{centre } (x_c, y_c)$
Grid – standard		$N, \Delta x, \Delta y, i, j, \theta, \text{start pt } (x_s, y_s)$
Grid – rectangle		$N, \Delta x, \Delta y, i, j, \theta, \text{start pt } (x_s, y_s)$
Corner Pattern		$N, \Delta x, \Delta y, i, j, \theta, \text{start pt } (x_s, y_s)$
Polar		$N, R, \Delta \phi, \theta, \text{start pt } (x, y), \text{centre } (x_c, y_c)$ for each centre (x_c, y_c) is common for all circle patterns

FIGURE 2.22 – Motifs principaux recherchés par Urbanic et Elmaraghy [108].

Synthèse

Dans la littérature, quelques méthodes proposent d'extraire les répétitions contenues dans un objet, manufacturé ou non. Elles ont toutes pour point commun de rechercher des motifs particuliers à partir de sous-ensembles d'éléments similaires. Notons également que ces motifs sont tous définis dans un plan 2D, comme lors de la modélisation CAO.

Cependant, ces méthodes sont pour la plupart conçues pour traiter des modèles sans imprécisions. Or, la similarité et le positionnement d'objets imprécis est plus complexe et peut générer des ambiguïtés de construction, notamment lorsque la complexité des pièces traitées augmente. Pour utiliser ces méthodes dans le cadre de nos travaux, il semble nécessaire de régulariser au préalable les paramètres des primitives extraites afin de gagner en précision.

2.3.3 Symétries

Mitra *et al.* [73] présentent un état de l'art sur la recherche de plans de symétrie. Les méthodes sont catégorisées et comparées en fonction du type de symétrie recherchée : globale ou partielle, exacte ou approximée, extrinsèque ou intrinsèque (en fonction d'une surface plane ou courbe, voir Fig. 2.23). Une symétrie globale correspond à une transformation de l'objet, autre que l'identité, permettant de retrouver le même objet. Une symétrie partielle est basée sur le même principe, mais seulement sur une sous-partie de l'objet. Par contre, une symétrie n'est jamais tout-à-fait exacte sur des objets réels, ce qui impose souvent la définition d'un système de tolérances. Enfin,

les symétries intrinsèques, qui ne dépendent pas de la pose de l'objet ou de l'être vivant, semblent plus adaptées d'après les auteurs (Fig. 2.23). Cependant, dans notre contexte, les objets manufacturés rigides ne contiennent que des symétries extrinsèques.



FIGURE 2.23 – Exemple de symétrie [73] : a) Intrinsèque et b) Extrinsèque.

Li *et al.* [62] proposent une méthode permettant de détecter les symétries au sein d'un modèle B-Rep (Fig. 2.24). Pour cela, ils commencent par décomposer l'objet sous forme d'arbre d'assemblage, à partir des informations déjà présentes dans le modèle B-Rep (Fig. 2.24.a). Ils cherchent ensuite les plans de symétrie au sein de chaque feuille de l'arbre obtenu, en fonction du type de solide correspondant (Fig. 2.24.c). Ensuite, les auteurs recherchent des répétitions constituées de plusieurs feuilles de l'arbre associées à des solides identiques (Fig. 2.24.d). Enfin, un parcours de l'arbre allant des feuilles vers la racine permet de retrouver des symétries de manière plus globale (Fig. 2.24.e).

Cependant, cette méthode est basée sur des modèles B-Rep, contenant déjà de nombreuses informations liées à l'assemblage. Dans le cadre de nos travaux, il semble difficile d'adapter cette méthode compte tenu de toutes les informations imprécises ou manquantes.

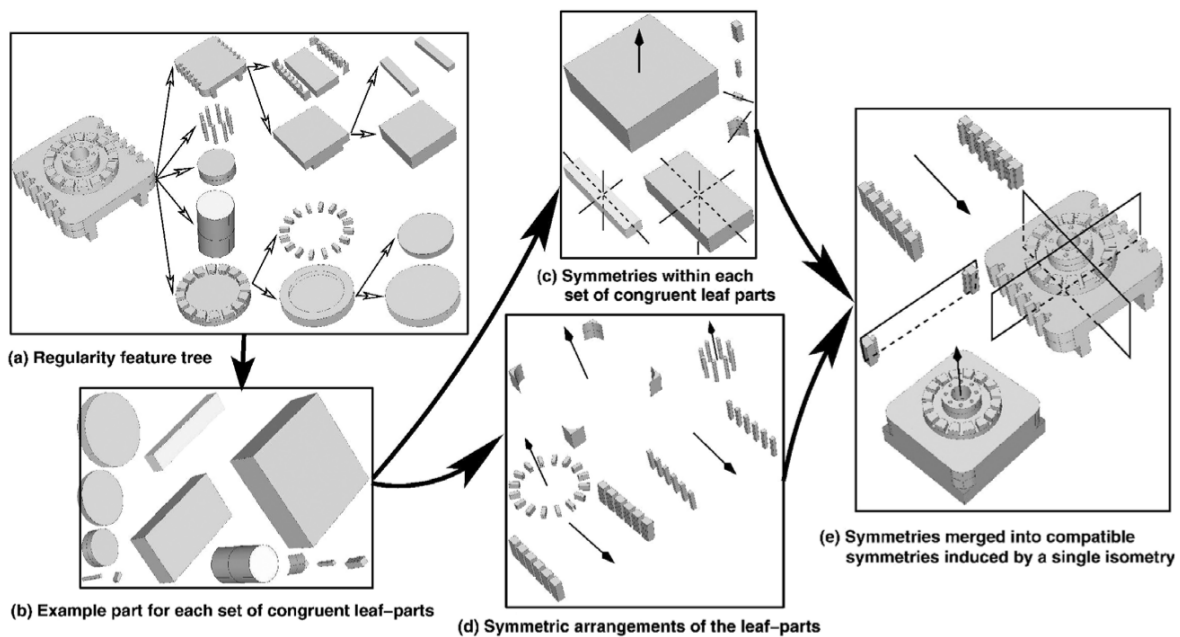


FIGURE 2.24 – Schéma global de la méthode proposée par Li *et al.* [62].

Li *et al.* [61] présentent une méthode de détection des symétries locales et globales d'un modèle B-Rep (Fig. 2.25). La première étape consiste à fusionner des primitives ou volumes concourants. Ensuite, un plan ou un axe de symétrie est calculé à partir d'une primitive et de ses voisines. Les primitives voisines sont alors itérativement agrégées, jusqu'à ne plus respecter la symétrie ou avoir couvert tout l'objet. Cette étape est réitérée pour chaque primitive avec des plans et axes différents. Afin d'obtenir une meilleure précision, celle-ci se base sur les équations des primitives pour détecter les symétries, et non sur une discrétisation ou des contours.

Cependant, cette méthode est conçue pour des modèles sans imprécisions. Par conséquent, des adaptations seraient nécessaires afin de pouvoir l'utiliser dans le cadre de nos travaux, à partir d'un ensemble de primitives imprécises. En particulier, des informations peuvent être manquantes lors d'une numérisation, fournissant au final des symétries partielles non prises en compte dans cette approche.

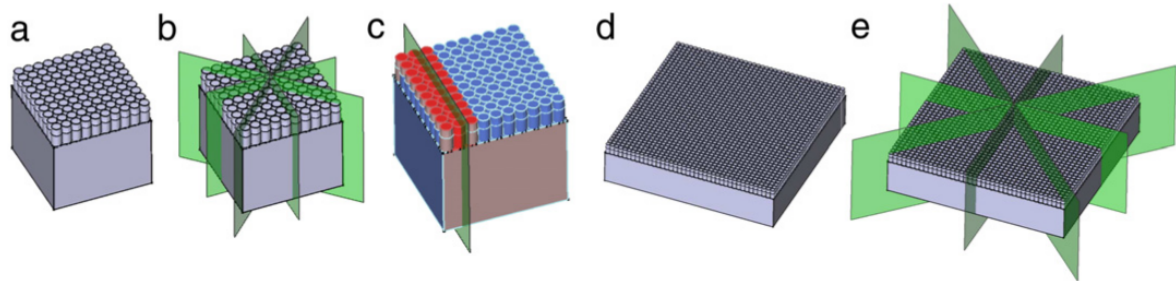


FIGURE 2.25 – Détection des symétries locales et globales de Li *et al.* [61].

Kovács *et al.* [54] proposent de détecter des contraintes de symétrie, mais cette fois-ci entre des primitives extraites à partir de maillages issus de numérisation. Dans la section 2.2, nous avons vu que les auteurs retrouvent des orientations dominantes grâce à une sphère Gaussienne. Une discrétisation de l'espace sous forme de grille 3D cubique est ensuite construite en fonction de ces orientations. Enfin, cette grille est analysée selon plusieurs angles de vue afin de retrouver des symétries partielles, à partir de plans de symétrie définis soit entre deux cases, soit au milieu d'une case. Pour cela, chaque case est au préalable catégorisée soit en intersection avec les primitives, soit vide. Une symétrie est alors détectée en comparant les catégories des cases de chaque côté du plan. Les auteurs proposent ainsi des symétries entre plusieurs primitives différentes et des auto-symétries (Fig. 2.26). Le point crucial de cette méthode est la recherche de symétries partielles. En effet, à cause des imprécisions, il n'existe pas de symétrie totale dans les maillages issus de numérisation.

Cependant, la grille permettant de détecter les symétries dépend de paramètres tels que sa résolution dans chaque direction. Or, il semble difficile de choisir ces paramètres pour des pièces complexes. De plus, les contours utilisés ici correspondent à ceux du maillage et non des primitives, qui représentent rarement les contours du modèle initial de manière fidèle.

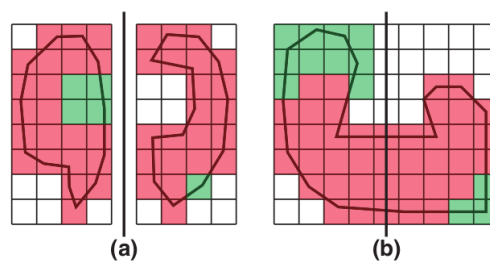


FIGURE 2.26 – Symétries partielles de Kovács *et al.* [54] : a) Entre deux primitives, b) Au sein d'une primitive.

Vilmart *et al.* [110] présentent une méthode de détection et catégorisation des symétries à partir d'un modèle B-Rep. La détection des plans de symétrie est basée sur les mêmes principes que la méthode de Li *et al.* [61]. Ensuite, les auteurs proposent de reconnaître des combinaisons particulières de plans de symétrie mettant en jeu les mêmes sous-ensembles de primitives. En effet, une structure géométrique peut fournir plusieurs plans de symétrie, souvent orthogonaux entre eux ou encore partageant un axe ou un point d'intersection commun (Fig. 2.27). Cela permet d'en déduire des informations intéressantes, telles que la construction d'un repère tridimensionnel (Fig. 2.27.c) ou d'une symétrie axiale discrète (Fig. 2.27.d). Ces symétries sont alors utilisées pour caractériser l'objet.

Cependant, cette méthode est conçue pour des modèles sans imprécisions. Celle-ci nécessite donc quelques adaptations pour être utilisée dans le contexte de nos travaux.

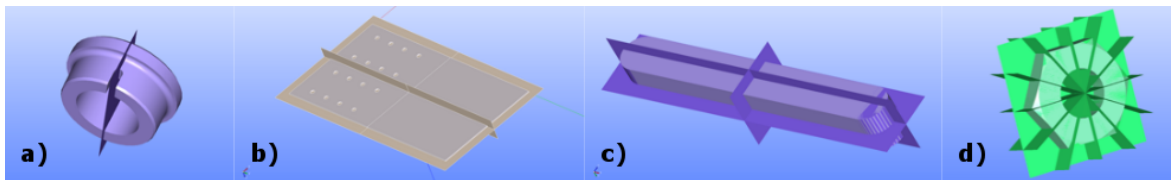


FIGURE 2.27 – Symétries de Vilmart *et al.* [110] : a) 1 plan, b) 2 plans, c) 3 plans, d) Axe discret à 6 plans.

Synthèse

La détection des symétries semble très performante dans le cas de modèles sans imprécisions. Cependant, dans le cas des reconstructions à partir de maillages issus de numérisation, il semble difficile d'appliquer directement les méthodes existantes. En effet, les symétries peuvent alors être seulement partielles, et donc plus difficilement détectables. De plus, les symétries extraites devraient se baser sur les contours du modèle, correspondant aux intersections entre les primitives. Or, dans le cas de la rétro-ingénierie, nous avons vu que ces contours souffrent également d'imprécisions. Il semble donc que les méthodes actuelles de détection des symétries ne soient pas adaptées à notre problématique.

2.4 Application à l'embellissement

Dans les sections précédentes, nous avons vu comment extraire des informations liées à la modélisation d'un objet à partir du modèle reconstruit. Ces informations peuvent alors être utilisées dans une dernière étape, afin d'améliorer la reconstruction et obtenir un modèle final plus *beau* ou respectant des contraintes particulières. Le terme d'embellissement (*beautification* en anglais) a premièrement été défini par Langbein [57] au cours de sa thèse. Il correspond à une étape de modification des paramètres des primitives, ayant pour objectif de "régulariser" géométriquement le modèle reconstruit. Ainsi, les primitives sont régularisées afin de respecter des relations géométriques telles que le parallélisme, l'orthogonalité ou la concentricité. Cela permet, par exemple, de corriger ou réparer un modèle ayant subi des distorsions pendant sa fabrication ou son utilisation, ou encore compenser les imprécisions des outils de mesure.

Dans la littérature, nous pouvons distinguer trois grandes classes de méthodes d'embellissement. La première approche consiste à extraire des relations géométriques entre paires de primitives puis à les régulariser. La seconde classe regroupe, quant à elle, les méthodes basées sur une analyse d'un sous-ensemble de primitives permettant de mieux structurer les relations obtenues. Enfin, la troisième approche consiste à analyser l'ensemble des primitives afin de déterminer des relations globales et ainsi régulariser plusieurs primitives en même temps.

Analyse de paires de primitives

Une première approche consiste à détecter des relations géométriques entre paires de primitives afin de régulariser leurs paramètres. La méthode d'embellissement proposée par Langbein *et al.* [56] construit un système de contraintes définies par de nombreuses relations géométriques approximativement présentes dans le modèle reconstruit. Par exemple, des angles entre des orientations ou des distances entre des positions peuvent être similaires entre plusieurs paires de primitives. Un algorithme itératif tente alors de trouver une solution au système en satisfaisant un maximum de contraintes. Pour cela, un sous-ensemble consistant de contraintes est sélectionné, puis de nouvelles contraintes sont ajoutées progressivement jusqu'à obtention d'un modèle suffisamment contraint.

Cependant, la construction d'un tel système peut être très complexe. En effet, les objets manufacturés, même relativement simples, contiennent souvent de très nombreuses relations géométriques (Fig. 2.28), dont certaines peuvent être redondantes. Par exemple, la relation de parallélisme est équivalente à une succession de deux orthogonalités, mais dans ce cas les trois

contraintes sont générées, ce qui complexifie la résolution du système. Enfin, de nombreuses vérifications sont nécessaires à chaque itération pour garantir la consistance du système en vérifiant qu'il n'y a aucune contradiction entre les contraintes. Cela implique souvent des temps de calcul importants, notamment pour les pièces complexes.

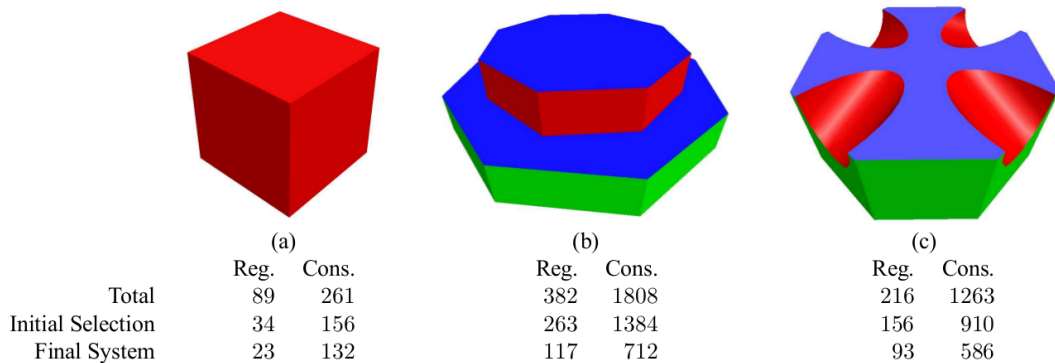


FIGURE 2.28 – Résultats de Langbein *et al.* [56] avec de nombreuses régularités et contraintes. Le système final contient moins de 50% des contraintes initiales, principalement à cause de leur redondance.

Chen et Feng [22] proposent d'utiliser des algorithmes de rétro-ingénierie afin *d'idéaliser* des maillages issus de numérisation (voir section 1.2). Dans leur approche, les primitives extraites sont régularisées afin de respecter des relations de parallélisme et d'orthogonalité avec le plus grand plan détecté. Ainsi, l'orientation de chaque plan approximativement parallèle ou orthogonal au plan de référence est modifiée afin de respecter exactement la relation correspondante.

Cependant, puisque ce plan de référence est extrait à partir des points du maillage, il est imprécis. Cette méthode implique donc une propagation d'erreur indésirable. De plus, les autres relations entre les paires de primitives ne contenant pas le plan de référence ne sont pas abordées, alors qu'elles sont elles aussi impactées par la régularisation. Par conséquent, cette approche n'est pas suffisamment générale.

Analyse de sous-ensembles de primitives

Une seconde approche consiste à analyser l'objet par sous-parties, puis d'embellir ces dernières indépendamment. Nous avons vu dans les sections 2.2 et 2.3 que Kovács *et al.* [54] proposent d'extraire des orientations principales et des symétries à partir d'une sphère Gaussienne et d'une grille 3D, illustrée dans la figure 2.29. Les auteurs proposent ensuite d'embellir les primitives à partir de relations d'orientation et de continuité. Dans leur méthode, un ajustement contraint des primitives est alors réalisé afin que celles-ci respectent des relations comme la tangence par exemple. Ce dernier est basé sur les travaux de Benkő *et al.* [9], détaillés dans la section 1.2.3.

Cependant, les orientations principales ne sont détectées qu'à partir des plans alors que les objets utilisés contiennent d'autres primitives. De plus, nous avons vu dans la section 2.3 que la construction de la grille 3D repose sur un paramétrage de l'utilisateur en fonction de la pièce (échelle, complexité...). Enfin, les primitives n'étant régularisées que localement en fonction de leurs voisins, il semble difficile de s'assurer d'un respect global des relations géométriques.



FIGURE 2.29 – Embellissement de Kovács *et al.* [54] à partir d'une grille 3D et de symétries partielles.

Analyse de l'ensemble des primitives

Une troisième approche consiste à analyser l'objet de manière plus globale en prenant en compte toutes les primitives. Oesau *et al.* [80] proposent de régulariser les plans durant l'extraction de primitives, à partir de relations globales de parallélisme et d'orthogonalité. Au départ, des points sont aléatoirement choisis comme graines, puis agrégés à leurs plus proches voisins. A chaque itération, ces sous-ensembles de points sont approximés par des plans, puis étendus au maximum en agrégeant les points proches du plan correspondant. Ensuite, un graphe des relations parallèles et orthogonales est construit à partir des paramètres d'orientation des plans. Des relations globales sont alors définies d'après le graphe, puis les plans sont régularisés en conséquence. Enfin, le processus est réitéré jusqu'à stabilité de l'ensemble des plans.

Cependant, l'approche ne prend en compte que les primitives planes et les relations de parallélisme et d'orthogonalité. Or, dans la majorité des objets manufacturés, nous pouvons trouver d'autres types de primitive, ainsi que des angles particuliers différents de 90° , par exemple 45° . De plus, ces relations, pourtant censées être globales, sont approximées lors des premières itérations à partir de plans peu représentatifs de l'objet car estimés à partir de très peu de points.

La méthode proposée par Li *et al.* [63] est associée à une approche de type RANSAC [34] (voir section 1.3), permettant d'ajouter itérativement de nouvelles primitives, de plus en plus petites et précises. Au départ, les primitives extraites sont analysées par paires pour définir des relations de parallélisme et d'orthogonalité, et ainsi construire un graphe d'orientation (Fig. 2.30). Ce graphe est ensuite simplifié en fusionnant les noeuds parallèles, de manière à ne conserver que les orientations principales. Ensuite, à chaque étape, le graphe est raffiné en fonction des nouvelles primitives ajoutées. Ces dernières sont alors régularisées en fonction des relations obtenues.

Cependant, un nombre croissant de primitives nécessite d'être régularisé à chaque étape, ce qui implique une complexité non-linéaire. De plus, les relations étant détectées par paires de primitives comme pour la première approche, une propagation d'erreur subsiste, en particulier lorsque les premières primitives extraites sont trop imprécises.

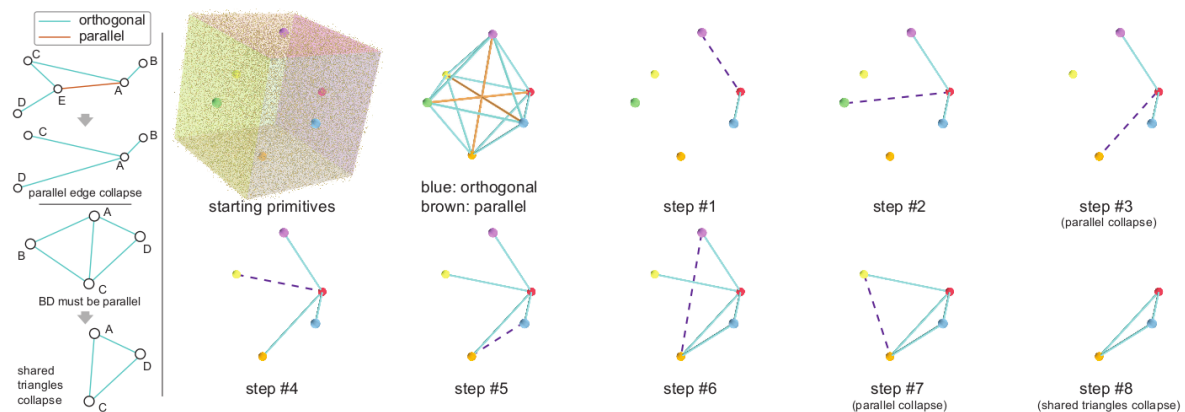


FIGURE 2.30 – Construction progressive d'un graphe d'orientation réduit de Li *et al.* [63].

Cas particulier de la réparation

Dans un autre registre, Gao *et al.* [38] présentent un algorithme d'embellissement permettant d'améliorer les contours des primitives définies dans un modèle B-Rep. Celui-ci a pour objectif de corriger des "erreurs topologiques" telles que des trous au milieu des primitives ou des chaînes de petites primitives identiques et adjacentes, sans modifier la géométrie de l'objet. Pour cela, les auteurs considèrent uniquement des erreurs isolées et localisées. La correction de certaines erreurs pouvant parfois amener à en créer d'autres, ces dernières sont corrigées successivement selon un ordre prédéfini (Fig. 2.31). Cela permet alors de s'assurer que le modèle final n'en contient plus. Cependant, cette méthode est conçue pour corriger des modèles B-Rep. Or, dans le cadre de nos travaux, nous n'avons qu'un ensemble de primitives indépendantes.

Sequence of topological repair											
Modification order	1	2	3	4	5	6	7	8	9	10	11
Fixing → can cause ↓	Face gaps	Edge gaps	Multiface gaps	Pinched faces	Small face chains	Sliver faces	Short edge chains	Adjacent faces	Small faces	Merging edges	Short edges
Face gaps	No	No	No	No	No	No	No	No	No	No	No
Edge gaps	No	No	No	No	No	No	No	No	No	No	No
Multiface gaps	No	No	No	No	No	No	No	No	No	No	No
Pinched faces	No	No	Yes	No	No	No	No	No	No	No	No
Small face chains	No	Yes	Yes	Yes	No	No	No	No	No	No	No
Sliver faces	No	No	Yes	Yes	No	No	No	No	No	No	No
Short edge chains	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
Adjacent faces	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No
Small faces	No	No	Yes	Yes	No	No	No	Yes	No	No	No
Merging edges	No	Yes	No	Yes	Yes	No	Yes	Yes	Yes	No	No
Short edges	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No

FIGURE 2.31 – Corrections des “erreurs topologiques” de manière ordonnée selon Gao *et al.* [38].

Synthèse

Dans la littérature, plusieurs méthodes proposent une étape d'embellissement permettant de régulariser les primitives de sorte qu'elles respectent des relations géométriques. Cependant, ces dernières ne prennent pas toujours formellement en compte les systèmes de référence, décrits dans la section 2.2, qui sont pourtant utilisés au cours de la modélisation et à l'origine même des relations géométriques entre les primitives. Il semble tout de même que les méthodes proposant une analyse globale des primitives permettent généralement une meilleure cohérence des relations géométriques détectées, en particulier lorsque les primitives sont imprécises. De plus, l'objectif des méthodes actuelles semble se limiter aux relations géométriques comme seul critère. En effet, dans le cas de la rétro-ingénierie, peu de méthodes proposent d'utiliser le maillage sous-jacent comme référence afin de limiter les écarts lors de l'embellissement. Or, l'objectif du processus de reconstruction est principalement de fournir un modèle correspondant aux données initiales. Il serait alors intéressant, dans le cadre de nos travaux, de proposer une méthode permettant de respecter les relations géométriques tout en garantissant la proximité entre le modèle reconstruit et le maillage initial.

2.5 Synthèse

La modélisation CAO d'une pièce est généralement guidée par un système de référence (repère orthonormé, plans, axes ou points) et des règles de construction permettant de créer des motifs particuliers (entités, répétitions ou symétries). Cela permet de simplifier la conception de pièces respectant des relations géométriques telles que le parallélisme, l'orthogonalité, la concentricité ou la tangence par exemple. Cependant, ces règles et références ne sont généralement pas utilisées dans les méthodes existantes de rétro-ingénierie. Pour reconstruire un modèle plus fidèle à l'original, il semble nécessaire de retrouver ces informations liées à la conception.

Un système de référence contient généralement un repère orthonormé ainsi que des plans, droites et points fixes permettant de contraindre la pièce dans sa totalité. Plusieurs méthodes proposent d'extraire certaines de ces références à partir d'un maillage ou d'un ensemble de primitives, en commençant souvent par une analyse des orientations. Cependant, ces dernières sont pour la plupart limitées à une analyse des orientations à partir d'un modèle B-Rep et ne prennent pas en compte toutes les primitives.

D'autres méthodes proposent d'extraire des motifs particuliers à partir des primitives. Parmi celles-ci, nous trouvons essentiellement des entités, des répétitions et des symétries. Cependant, ces dernières partent souvent de modèles B-Rep, qui contiennent des informations structurelles en plus de l'ensemble des primitives.

Dans le cadre de la rétro-ingénierie, il semble nécessaire d'appliquer une étape d'embellissement permettant de régulariser les paramètres des primitives. Cependant, les méthodes existantes ne semblent pas assez performantes pour reconstruire un modèle suffisamment précis.

Il serait alors intéressant de concevoir une méthode permettant d'embellir un ensemble de primitives en s'aidant d'autres informations telles qu'un système de référence et des motifs particuliers. La méthode d'analyse et d'embellissement présentée au chapitre 4 tente de répondre à cette problématique. Nous montrons ainsi que le système de référence utilisé lors de la modélisation peut être déduit à partir des primitives, et permet alors de régulariser les paramètres de ces dernières. Par la suite, nous proposons également d'utiliser une détection de motifs particuliers afin d'améliorer le positionnement des primitives.

Deuxième partie

Contributions



Chapitre 3

Procédure robuste d'extraction de primitives dans des maillages 3D numérisés

« Si debugger, c'est supprimer des bugs, alors programmer ne peut être que les ajouter. »

Edsger Dijkstra

Résumé

Comme nous l'avons vu au chapitre 1, l'extraction de primitives à partir de maillages 3D peut s'effectuer suivant deux approches : les méthodes incrémentales et les méthodes directes. Il semble que l'approche incrémentale soit la plus adaptée à notre problématique. Cependant, les algorithmes actuels ne semblent pas suffisamment robustes pour garantir la fiabilité des primitives dans le cas de maillages issus de numérisation, qui contiennent généralement de nombreuses imprécisions. Nous proposons dans ce chapitre une nouvelle méthode incrémentale composée de trois étapes.

Après une brève introduction (section 3.1), nous proposons dans la section 3.2 une segmentation hiérarchique de maillages 3D à partir de l'analyse globale des distributions de courbures. Nous détaillons ensuite l'ajustement des primitives sur les sous-maillages dans la section 3.3, puis une seconde étape de segmentation dans la section 3.4. Dans la section 3.5, nous décrivons une analyse plus locale des distributions de courbures qui permet d'adapter automatiquement les paramètres et les tolérances de calcul des primitives à partir des caractéristiques du maillage sous-jacent. Enfin, nous présentons des résultats expérimentaux (section 3.6), suivis d'une conclusion (section 3.7).

Sommaire

3.1 Introduction	64
3.2 Segmentation "arêtes vives"	65
3.2.1 Analyse de la distribution de courbure	65
3.2.2 Extraction des sous-maillages de forme homogène	68
3.2.3 Extraction récursive des arêtes vives	71
3.2.4 Résultats expérimentaux	72
3.3 Ajustement global des primitives	76
3.3.1 Optimisation d'une primitive sur un maillage	76
3.3.2 Identification du type de primitive	78
3.4 Segmentation "arêtes lisses"	80
3.5 Adaptation automatique de l'ajustement local des primitives	82
3.5.1 Méthode d'ajustement local de primitives à partir d'un sous-maillage	82
3.5.2 Identification du type de primitive majoritaire	83
3.5.3 Calcul automatique des tolérances	84
3.5.4 Exemple d'application	86
3.6 Résultats expérimentaux	87
3.7 Conclusion	90

3.1 Introduction

Afin de répondre aux problématiques mises en avant au chapitre 1, nous proposons une nouvelle approche d'extraction de primitives inspirée des méthodes incrémentales. La figure 3.1 en illustre les différentes étapes.

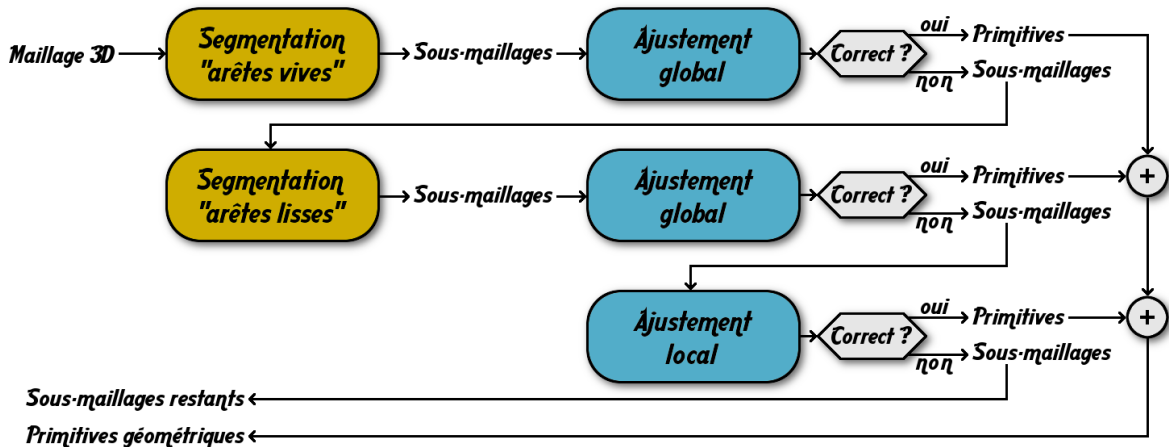


FIGURE 3.1 – Schéma global de notre approche d'extraction de primitives.

Premièrement, le maillage est segmenté grâce à une analyse des distributions des courbures. L'objectif est, grâce à cette analyse, de discriminer les points du maillage appartenant aux arêtes “vives” de l'objet. Ensuite, des sous-maillages sont extraits par croissance de régions. Cette segmentation est appliquée de manière récursive, permettant alors d'éliminer des arêtes plus ou moins vives. Cette étape est détaillée dans la section 3.2.

Deuxièmement, chaque sous-maillage est analysé afin d'approximer une primitive géométrique de manière globale. Pour cela, nous utilisons une combinaison de régressions qui permet d'estimer les paramètres des primitives à partir des points des sous-maillages. Cependant, ces paramètres ne sont pas toujours précis (voir section 1.2.3). Pour pallier à ce problème, une optimisation des paramètres permet de minimiser les distances entre les points et la surface de la primitive correspondante. Cette étape est présentée dans la section 3.3.

Troisièmement, lorsqu'aucune primitive ne peut être approximée correctement à partir d'un sous-maillage, ce dernier est segmenté à nouveau à l'aide d'une seconde analyse des distributions des courbures. En effet, certains sous-maillages peuvent comporter plusieurs primitives qui ne sont pas séparées par des arêtes vives. Cela suggère généralement des primitives tangentes. L'objectif est alors cette fois-ci de détecter les arêtes “lisses”. Cette étape est décrite dans la section 3.4.

Quatrièmement, une seconde étape d'ajustement global des primitives est réalisée à partir des sous-maillages obtenus après la seconde segmentation. Cependant, certains sous-maillages, notamment lorsqu'ils sont très imprécis, peuvent encore contenir plusieurs primitives.

Cinquièmement, un ajustement des primitives est réalisé localement sur les sous-maillages restants. Celui-ci procède par croissance de régions, à partir des valeurs des courbures en chaque point. Pour cela, un certain nombre de tolérances sont définies, par exemple pour détecter la similarité de deux valeurs de courbure. Une procédure automatique de calcul de ces tolérances, en fonction des sous-maillages et de leurs distributions de courbures, est alors réalisée. Cette dernière permet de détecter le type de primitive majoritaire et d'adapter les tolérances en fonction de la répartition des imprécisions au sein de chaque sous-maillage. Cette étape est détaillée dans la section 3.5.

3.2 Segmentation "arêtes vives"

Notre méthode de segmentation a pour objectif de séparer les différentes zones de l'objet correspondant aux primitives géométriques. Nous proposons alors d'extraire des sous-maillages de forme homogène, aussi appelés *uniformes*, qui contiennent chacun une seule primitive. Pour cela, nous nous basons sur une analyse des courbures pour caractériser la forme de manière robuste et ainsi détecter les arêtes vives d'un objet (Fig. 3.2).

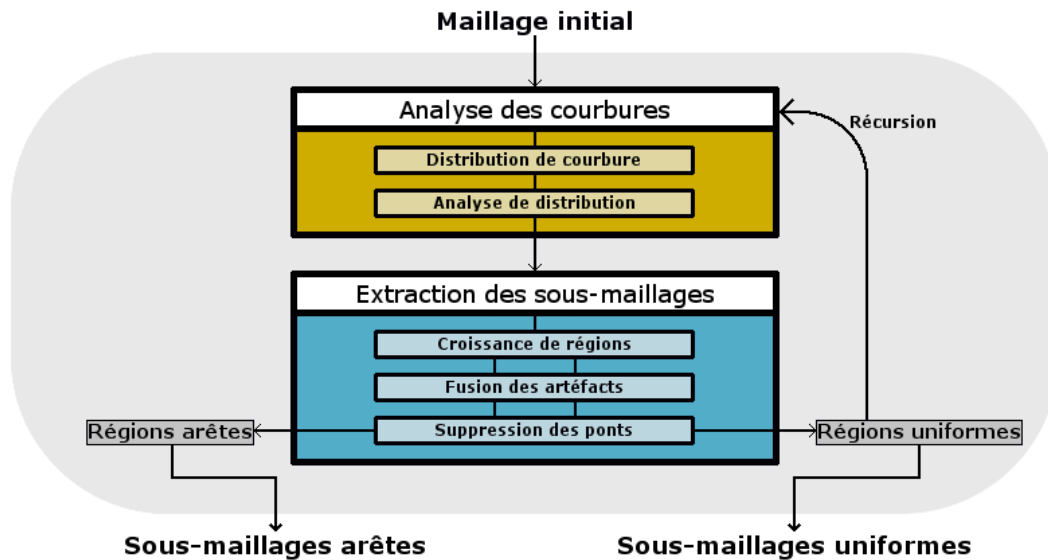


FIGURE 3.2 – Vue d'ensemble de la segmentation "arêtes vives".

Nous commençons par calculer la courbure sur le maillage 3D. Ensuite, nous en analysons la distribution afin de catégoriser chaque point du maillage selon qu'il appartient à une arête vive de l'objet ou à une zone de forme homogène. Cette catégorisation permet de construire des sous-maillages à l'aide d'une croissance de régions. Enfin, nous proposons deux étapes de post-traitement permettant d'améliorer les résultats : la fusion des artéfacts et la suppression des ponts. L'algorithme de segmentation est alors appliqué récursivement sur les sous-maillages *uniformes* afin de correctement extraire toutes les arêtes vives de l'objet.

3.2.1 Analyse de la distribution de courbure

Tout d'abord, les courbures principales sont calculées en chaque point du maillage avec la méthode de Bénéière *et al.* [5], décrite dans la section 1.2.2. Nous construisons alors un histogramme de la courbure moyenne signée. Le choix d'utiliser la courbure moyenne a été dirigé par ses propriétés intéressantes de différenciation des primitives (voir tableau 3.1). En effet, un cylindre a par exemple une des deux courbures principales nulle pouvant être confondue avec celle d'un plan, alors que la courbure moyenne ne peut pas être nulle pour un cylindre. De plus, le signe de la courbure moyenne donne une tendance de convexité locale et permet ainsi de différencier les zones convexes des zones concaves.

Primitive	Courbures			
	Min	Max	Moyenne	Gaussienne
Plan	0	0	0	0
Cylindre	0	$\frac{1}{rayon}$	$\frac{1}{2 \times rayon}$	0
Cône	0	variable	variable	0
Sphère	$\frac{1}{rayon}$	$\frac{1}{rayon}$	$\frac{1}{rayon}$	$\frac{1}{rayon^2}$

TABLEAU 3.1 – Valeurs des différentes courbures associées aux primitives géométriques.

Cependant, la construction d'un tel histogramme est confrontée à trois problématiques.

Premièrement, les valeurs de courbure sont inversement proportionnelles à l'échelle de l'objet. Cela ne permet pas de comparer les distributions obtenues à partir de maillages de différentes tailles. Nous proposons donc de normaliser les valeurs de courbure calculées, par rapport à la longueur moyenne \mathcal{L} des arêtes du maillage :

$$rayon_{normalisé} = \frac{rayon}{\mathcal{L}} \quad \text{d'où} \quad k_{normalisée} = \mathcal{L} \times k_{moyenne}. \quad (3.1)$$

En effet, les longueurs des arêtes reliant les points d'un maillage issu de numérisation peuvent être considérées en général assez homogènes. La distribution de courbure est ainsi indépendante des changements d'échelle.

Deuxièmement, chaque maillage fournit une plage de valeurs de courbure moyenne différente, en fonction de sa forme locale. Afin de concevoir un algorithme générique applicable à tout maillage, nous devons fixer une plage de valeurs de manière à homogénéiser la construction des histogrammes.

Supposons que toutes les arêtes du maillage soient de même longueur \mathcal{L} . La figure 3.3.a illustre les deux configurations extrêmes possibles avec un voisinage de 1. Lorsque les arêtes du maillage sont mises à plat, la courbure est nulle. Lorsqu'elles sont complètement repliées sur elles-mêmes, la courbure est égale à $\frac{2}{\mathcal{L}}$. Avec un voisinage de 2, par construction, nous obtenons la même valeur minimum mais une valeur maximum pouvant varier de $\frac{1}{\mathcal{L}}$ à $\frac{2}{\mathcal{L}}$ selon le nombre de points de chaque voisinage (Fig. 3.3.b). Nous en déduisons donc que les valeurs de courbure devraient être situées dans l'intervalle $[-\frac{2}{\mathcal{L}}; \frac{2}{\mathcal{L}}]$, soit l'intervalle $[-2; 2]$ après normalisation. Nous proposons donc de limiter les valeurs de courbure moyenne normalisée à cet intervalle.

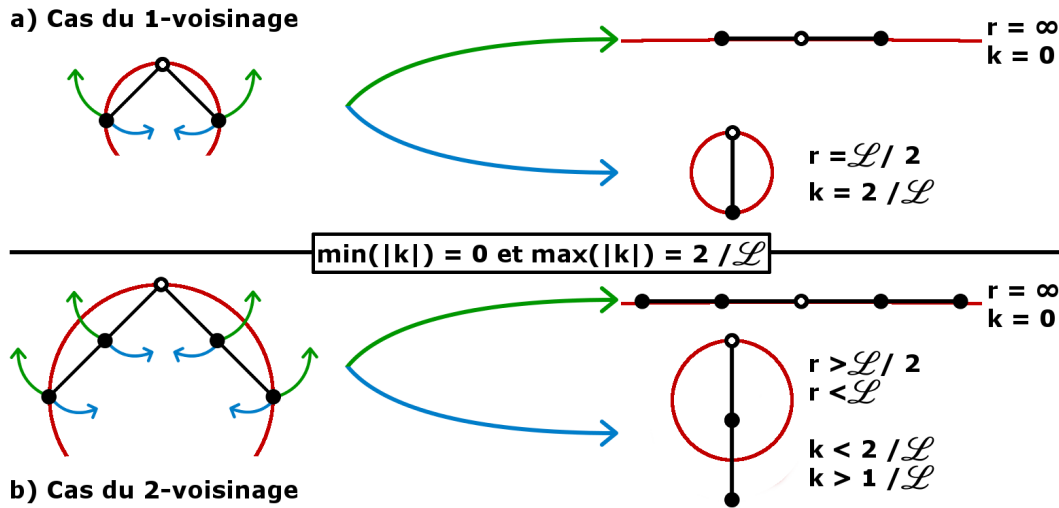


FIGURE 3.3 – Plage de valeurs des courbures absolues k en fonction de la longueur d'arête \mathcal{L} .

Troisièmement, l'échantillonnage des données impacte largement la construction d'un histogramme, comme nous l'avons vu dans la section 1.2.2. Pour pallier à ce problème, nous proposons d'approximer l'histogramme de manière continue pour chaque valeur centrale de classe x , grâce à une estimation par noyau définie par :

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i), \quad (3.2)$$

avec K_h la fonction noyau, n le nombre d'échantillons x_i et \hat{f}_h l'estimation de l'effectif de la classe.

Cependant, il est nécessaire de bien choisir le noyau. Prenons pour cela l'exemple d'une sphère. Lorsque la sphère est discrétisée avec une grande précision, toutes les valeurs de courbure sont identiques et égales à $\frac{1}{rayon}$ (Fig. 3.4.a). Or, dans la littérature, la plupart des approches supposent que les imprécisions de numérisation suivent une loi Gaussienne. Par exemple, rappelons que Masuda *et al.* [70] mesurent un bruit décrit comme Gaussien à partir de scanners laser. Lorsqu'un bruit Gaussien est ajouté artificiellement à cette sphère, nous remarquons que la distribution de courbure moyenne semble suivre une loi Gaussienne (Fig. 3.4.b-c).

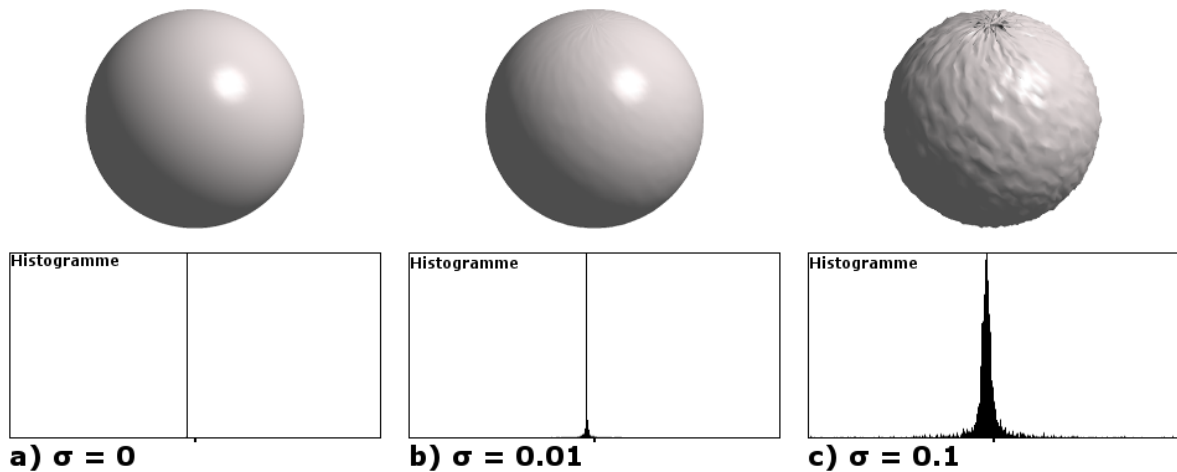


FIGURE 3.4 – Répartition des valeurs de courbure moyenne sur une sphère : a) Précise, b-c) Bruitée.

Dans ce sens, nous utilisons un noyau Gaussien, paramétré par un écart-type h :

$$K_h(x - x_i) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-x_i}{h}\right)^2}. \quad (3.3)$$

Ainsi, une loi Gaussienne est estimée pour chaque valeur de courbure à partir des valeurs proches. La forme générale de l'histogramme obtenu est alors moins sensible à l'échantillonnage choisi. Il est également à noter qu'un nombre impair de classes permet de rassembler les valeurs autour de zéro, correspondant aux points plans, dans la classe centrale.

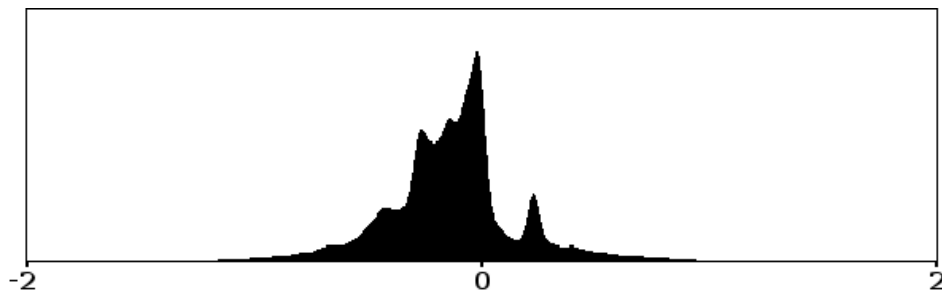


FIGURE 3.5 – Exemple d'histogramme de courbure moyenne signée normalisée.

Nous pouvons alors supposer que chaque primitive peut être représentée par un mode dans l'histogramme de courbure. En effet, chaque mode représente un intervalle de courbure homogène, correspondant généralement à un ensemble de points appartenant à la même primitive ou à des primitives similaires. Or, le rôle de la segmentation est de séparer ces primitives, et donc les modes de l'histogramme. Nous cherchons alors à définir des valeurs de seuils permettant de séparer ces intervalles. Finalement, l'analyse des courbures proposée consiste à détecter les pics, correspondant aux modes, et les vallées, correspondant aux limites des intervalles de courbures homogènes (Fig. 3.6).

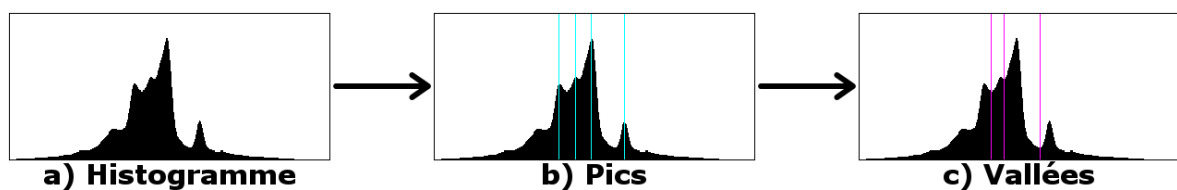


FIGURE 3.6 – Analyse de l'histogramme par recherche des extrema locaux.

La principale difficulté ici réside dans le chevauchement des modes représentant les primitives. Par exemple, les valeurs de courbure d'un cylindre contenant de nombreux points peuvent

masquer en grande partie celles d'un cylindre ayant moins de points et avec un rayon proche. Les primitives n'engendrent alors pas toujours des modes bien marqués dans l'histogramme (Fig. 3.7). Il n'est alors pas suffisant de rechercher directement les extrema locaux.

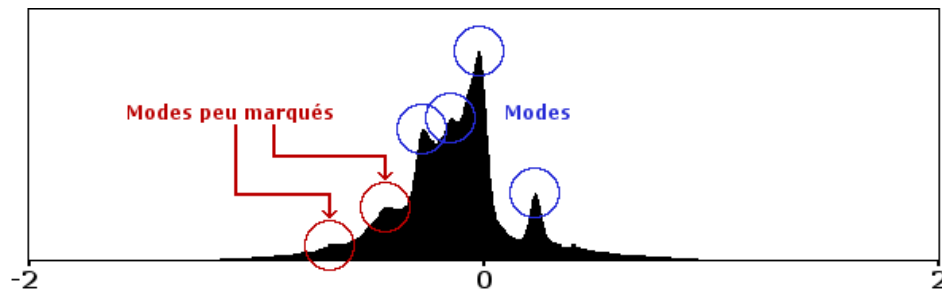


FIGURE 3.7 – Problème de chevauchement des modes.

Pour les séparer efficacement, nous pouvons retrouver les points d'inflexion des lois Gaussiennes correspondantes. Nous proposons pour cela un algorithme en quatre étapes.

1. Dériver deux fois le noyau Gaussien K_h .
2. Construire un nouvel histogramme à partir des valeurs de courbure et du noyau dérivé.
3. Rechercher les minima et maxima locaux grâce à une fenêtre glissante.
4. Supprimer les extrema non significatifs.

Afin d'éviter les erreurs dues aux fluctuations faibles, nous fixons un seuil de détection des extrema en fonction de l'écart-type des ordonnées de l'histogramme dérivé. Ainsi, un minimum ou maximum local ayant une valeur très proche de zéro n'est pas pris en compte. La figure 3.8 montre les résultats obtenus pour l'histogramme de la figure 3.7.

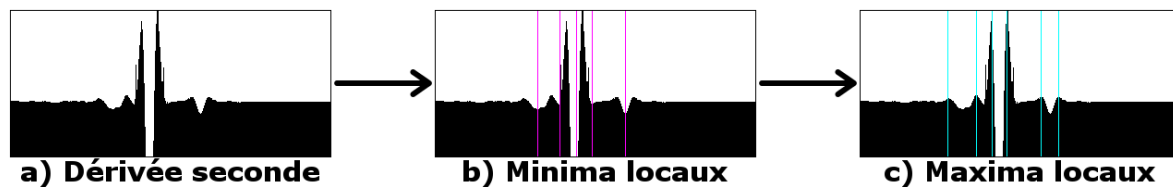


FIGURE 3.8 – Recherche des points d'inflexion dans un histogramme de courbure.

Les minima locaux obtenus correspondent alors aux pics dans l'histogramme original, et les maxima locaux définissent les vallées (Fig. 3.9). Nous remarquons que les modes peu marqués sont correctement séparés avec cette méthode.

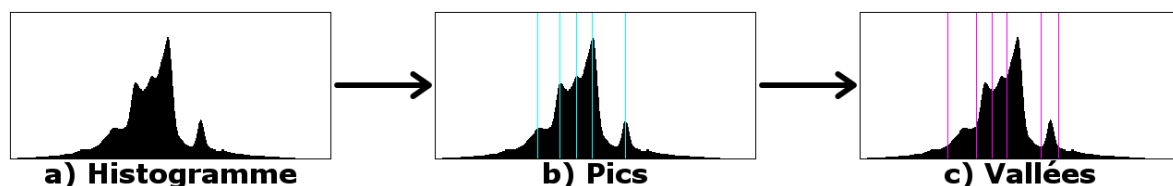


FIGURE 3.9 – Détection des pics et des vallées à l'aide de la dérivée seconde (Fig. 3.8).

3.2.2 Extraction des sous-maillages de forme homogène

La première partie de la segmentation fonctionne par extraction des arêtes vives de l'objet, correspondant aux sous-ensembles de points ayant des valeurs de courbure moyenne extrêmes. Pour extraire ces points, nous définissons deux seuils de courbure T_1 et T_2 , correspondant aux deux vallées limites de l'histogramme V_1 et V_k (Fig. 3.10). Ainsi, tout point dont la valeur de courbure est inférieure à T_1 ou supérieure à T_2 est considéré comme appartenant à une arête vive.

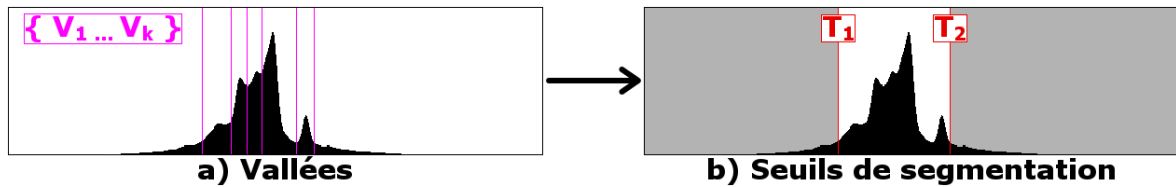


FIGURE 3.10 – Seuils de courbure permettant de détecter les arêtes vives (zone grise).

Ces deux seuils permettent alors de catégoriser chaque point du maillage selon s'il appartient à une arête vive de l'objet ou à une zone *uniforme*. La catégorisation des points peut ensuite être étendue aux triangles, en prenant la catégorie majoritaire de leurs points. Ainsi, les triangles *uniformes* sont les triangles dont au moins deux points ont une courbure située entre les deux seuils. Les autres triangles sont considérés comme appartenant aux arêtes vives (Fig. 3.11).

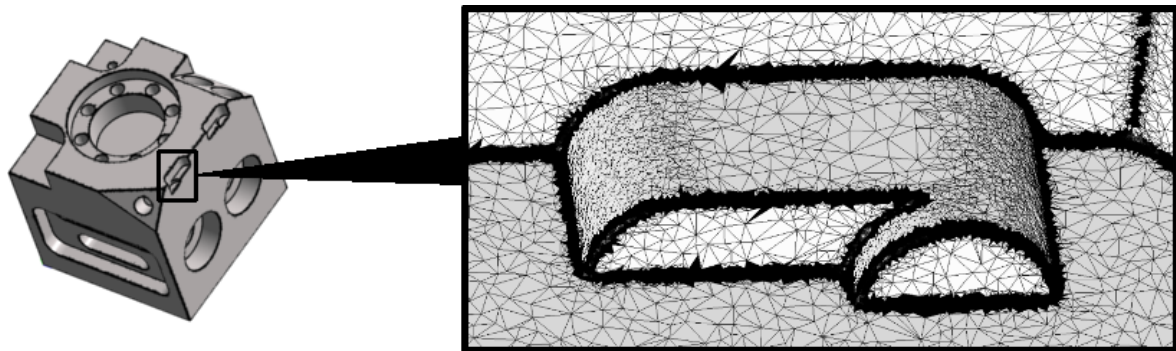


FIGURE 3.11 – Catégorisation des triangles d'un maillage numérisé par seuillage de la courbure moyenne : les triangles *arêtes vives* sont représentés en noir et les triangles *uniformes* en blanc.

Nous pouvons ensuite extraire les composantes connexes par croissance de régions. Celle-ci consiste à agglomérer les triangles voisins et de même catégorie entre eux. Cette croissance de régions est aussi appliquée aux triangles *arêtes*. La figure 3.12 illustre des étapes clés de cet algorithme, qui permet au final de définir des sous-maillages de forme homogène. Nous avons vu dans la section 1.2.1 que la catégorisation permet de s'abstraire du problème de choix de la graine. Dans notre cas, nous pouvons alors choisir arbitrairement les triangles graines.

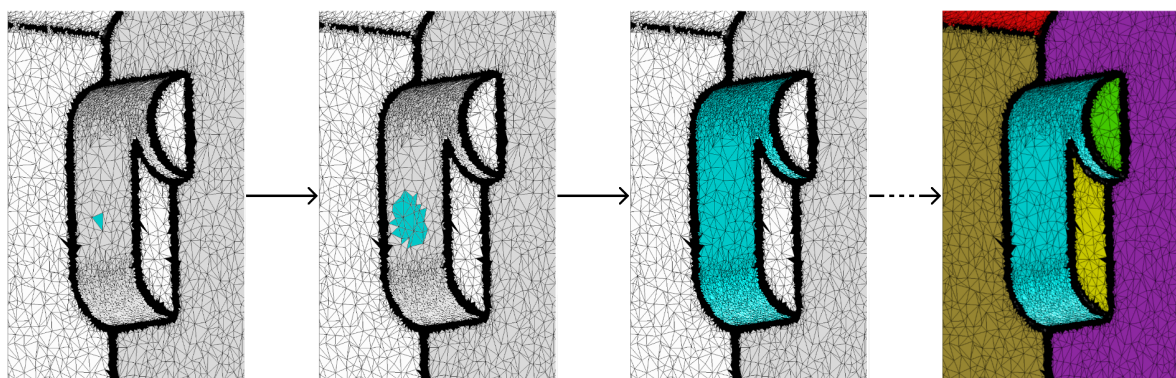


FIGURE 3.12 – Processus de croissance de régions après catégorisation des triangles.

Nous obtenons ainsi deux ensembles de sous-maillages *arêtes* et de sous-maillages *uniformes*. Cependant, à cause de l'imprécision des maillages numérisés, la catégorisation de certains points est incorrecte. Par conséquent, la croissance de régions peut engendrer une sur-segmentation ou une sous-segmentation. Nous réalisons alors deux étapes de raffinement des régions obtenues. De plus, il semble difficile de garantir l'extraction correcte de toutes les arêtes vives d'un maillage numérisé en une seule analyse de courbure. Nous proposons dans ce cas d'extraire les arêtes vives de manière récursive.

3.2.2.1 Correction de la sur-segmentation

Une sur-segmentation implique la construction d'artéfacts, c'est-à-dire des régions de quelques triangles n'ayant aucune signification du point de vue de l'objet. Par exemple, il semble fréquent d'obtenir de petites régions uniformes au milieu des "arêtes vives" d'un maillage imprécis (Fig. 3.13). Ces artéfacts sont principalement dûs aux imprécisions locales ne permettant pas de calculer de bonnes valeurs de courbure en tout point. Nous définissons ici un artéfact comme une région dont l'aire est inférieure à un seuil défini à partir de l'aire moyenne des triangles du maillage :

$$\text{Seuil} = \lambda \times \frac{1}{n} \sum_{i=1}^n \text{Aire}(T_i), \quad (3.4)$$

avec n le nombre de triangles T_i du maillage et λ un coefficient de taille qu'il est nécessaire d'adapter en fonction du type de maillage. Dans le cas de maillages numérisés relativement homogènes, $\lambda = 20$ semble en général suffisant. En quelques sortes, cela suppose que les régions de moins de 20 triangles ne peuvent pas correspondre à un sous-maillage support de primitives.

Si un artéfact n'est voisin que d'une région, alors il est possible de le fusionner avec cette dernière sans ambiguïté. Par contre, si un artéfact correspond à une région arête qui sépare plusieurs régions uniformes, il est possible que celui-ci soit tout de même pertinent. Les artéfacts ayant plusieurs régions voisines ne sont alors pas fusionnés pour ne pas modifier la connectivité globale des régions. Comme nous pouvons le voir dans la figure 3.13, la plupart des artéfacts n'ont qu'une seule région voisine et peuvent être fusionnés.

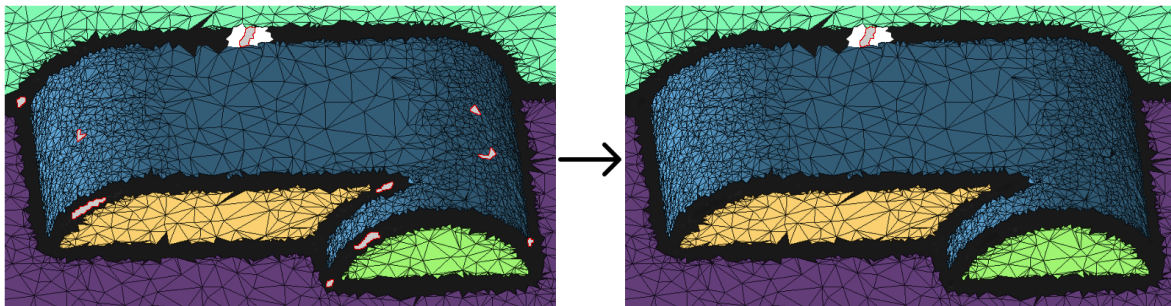


FIGURE 3.13 – Fusion des artéfacts (régions en gris et rouge) n'ayant qu'un seul voisin.

3.2.2.2 Correction de la sous-segmentation

Au contraire, une sous-segmentation implique la construction d'une grande région au lieu de plusieurs plus petites. La région peut alors contenir des *ponts*, qui correspondent à une sorte de goulet d'étranglement formé de quelques triangles seulement. L'objectif ici est donc de retirer ces *ponts*, afin de retrouver de grandes régions séparées. Pour cela, nous appliquons un processus d'ouverture, qui consiste à réaliser successivement une érosion et une dilatation des régions. L'érosion consiste à retirer les triangles d'une région s'ils sont situés sur sa frontière. La dilatation, au contraire, consiste à agglomérer les triangles adjacents à une région, c'est-à-dire s'ils touchent sa frontière. Notons que les ponts de largeur l peuvent être érodés en $\frac{l}{2}$ itérations (Fig. 3.14).

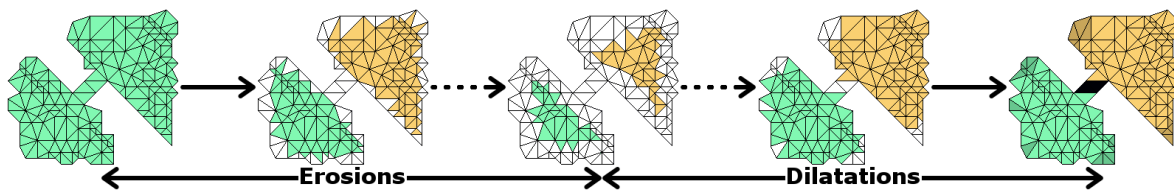


FIGURE 3.14 – Suppression des ponts entre régions.

La difficulté ici est de contrôler la quantité d'érosions afin de ne pas retirer de région pertinente. Pour cela, nous définissons un algorithme basé sur un degré d'érosion k_{max} (Algorithme 4).

Algorithme 4 Suppression des ponts par ouverture

Soit un ensemble de régions \mathcal{R} et un ensemble de ponts \mathcal{P}

pour k de 1 à k_{max} **faire**

pour tout région $R_i \in \mathcal{R}$ et l'ensemble de triangles F_i définissant sa frontière **faire**

si $Aire(F_i) \leq \lambda \times Aire(R_i)$ **alors**

pour tout triangle T non érodé de F_i **faire**

 Marquer T comme érodé au degré k

fin pour

fin si

fin pour

$\mathcal{R} \leftarrow$ Croissance de régions sur les triangles non-érodés

fin pour

pour k de k_{max} à 1 **faire**

pour tout triangle T érodé au degré k et voisin d'une région R_i **faire**

 Marquer T comme non érodé

fin pour

$\mathcal{R} \leftarrow$ Croissance de régions sur les triangles non-érodés

fin pour

$\mathcal{P} \leftarrow$ Croissance de régions sur les triangles érodés

$\mathcal{R} \leftarrow$ Régions après fusion des ponts P_i n'ayant qu'une seule région voisine R_j

renvoyer \mathcal{R}

Le degré d'érosion k_{max} permet de limiter la largeur maximale des ponts détectés. A chaque itération, nous mesurons la proportion de triangles érodés. Dans le cas où cette quantité serait trop grande, par exemple supérieure à 10% de la région, l'érosion n'est pas réalisée. Ce critère d'arrêt permet de ne pas supprimer de région pertinente. Une dilatation inverse est ensuite réalisée afin de reconstruire au mieux les régions initiales. Enfin, les triangles toujours marqués comme érodés à la fin du processus constituent les ponts. Ceux n'ayant qu'une seule région voisine sont alors fusionnés avec celle-ci, puisque non pertinents. Les régions finales, correctement séparées, sont conservées en tant que sous-maillages résultants.

3.2.3 Extraction récursive des arêtes vives

Afin de pouvoir traiter efficacement des maillages variés tout en garantissant la fiabilité des résultats obtenus, il semble nécessaire d'effectuer une récursion. En effet, il peut arriver que la première itération de segmentation ne suffise pas à découper l'objet assez finement. Ainsi, chaque itération a pour rôle d'affiner la segmentation (Fig. 3.15).

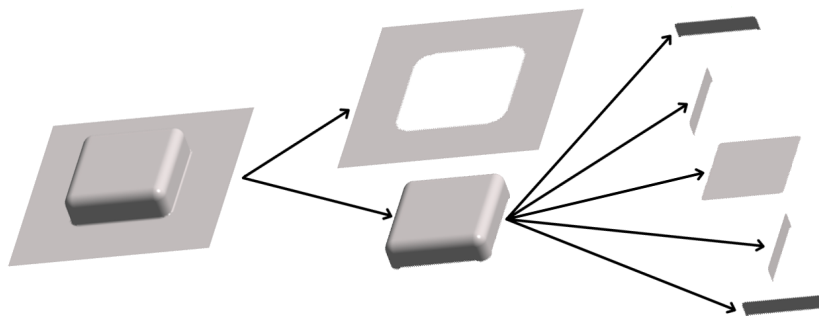


FIGURE 3.15 – Segmentation par extraction récursive des arêtes vives.

Le processus de segmentation décrit précédemment fournit un ensemble de sous-maillages. Une nouvelle distribution de courbure moyenne est alors construite pour chaque sous-maillage indépendamment, puis analysée pour en déduire de nouveaux sous-maillages. La condition d'arrêt de cette récursion est d'obtenir un seul sous-maillage après segmentation, ce qui, dans le pire des cas, correspond à un seul triangle. Cette condition est donc toujours vérifiée au pire au bout de n itérations, où n correspond au nombre de triangles du maillage initial.

3.2.4 Résultats expérimentaux

Notre méthode de segmentation a été intégrée à la chaîne de traitement développée par Bénière *et al.* [8]. Elle permet de grandement améliorer les résultats d'extraction de primitives, que ce soit en termes de qualité ou de rapidité. En effet, la segmentation permet d'obtenir des sous-maillages de forme homogène à partir desquels le calcul de la courbure n'est pas faussé (voir section 1.1). De plus, le traitement par sous-maillage permet par la suite d'ajuster des primitives sur des sous-maillages plus réduits, et ainsi améliorer les temps de calculs.

Nous avons testé notre segmentation et son impact sur l'extraction de primitives, sur de nombreux maillages issus de numérisation 3D. Tout d'abord, une étude qualitative est réalisée en analysant visuellement trois maillages segmentés. Ensuite, une étude plus quantitative permet de généraliser cette analyse à de nombreux maillages en évaluant les performances de l'algorithme. Dans le cadre de la rétro-ingénierie, l'objectif de cette segmentation est d'obtenir un sous-maillage par primitive. Dans ce sens, nous avons évalué manuellement le nombre de sous-maillages obtenus qui ne semblent contenir qu'une seule primitive.

Dans ces expériences, nous avons utilisé les paramètres suivants.

- Nous construisons des histogrammes de courbure contenant 4001 classes. Rappelons qu'un nombre impair de classe permet de regrouper les courbures planes. Puisque les valeurs de courbure sont normalisées et définies dans l'intervalle $[-2; 2]$, chaque classe représente un intervalle de courbure moyenne normalisée de taille 0.001.
- Le noyau Gaussien utilisé est paramétré avec un écart-type $h = 0.01$. Expérimentalement, cette valeur semble être un bon compromis entre la précision et le chevauchement des différentes lois Gaussiennes, ce qui permet de reconstruire une distribution de manière fidèle.

3.2.4.1 Etude qualitative

Cette section présente les résultats de notre algorithme de segmentation sur les trois maillages mécaniques de la figure 3.16. Ces maillages ont des dimensions et résolutions différentes, ce qui permet de tester la robustesse de la segmentation. Pour chaque maillage, nous présentons les arêtes vives détectées lors de la segmentation, ainsi que l'ensemble final de sous-maillages obtenus.

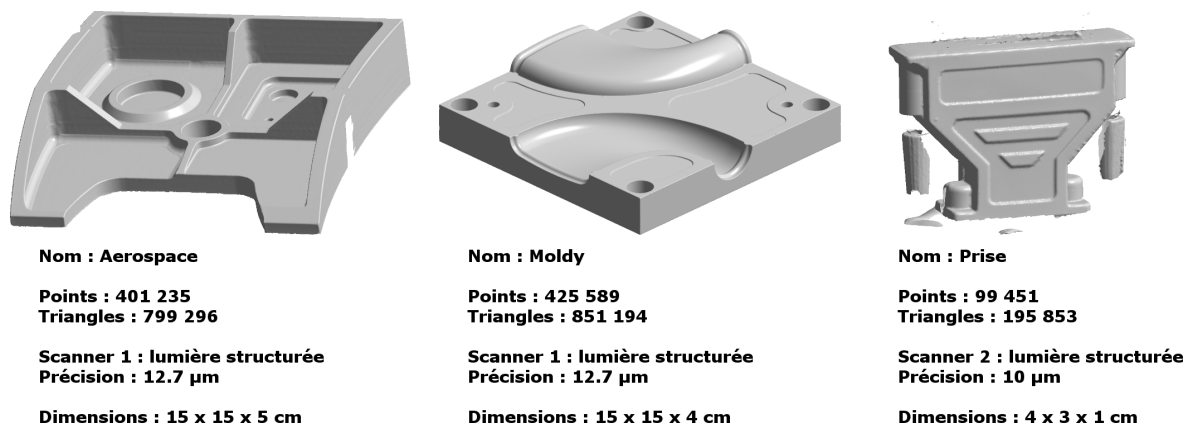


FIGURE 3.16 – Exemples de maillages 3D provenant de deux scanners différents.

La figure 3.17 illustre les résultats obtenus par segmentation de *Aerospace*. Les arêtes vives semblent globalement bien détectées (Fig. 3.17.a). Nous remarquons également que certains congés cylindriques et sphériques sont partiellement détectés comme arêtes vives. La segmentation récursive nous fournit au final un ensemble de 42 sous-maillages uniformes (Fig. 3.17.b). Parmi ceux-ci, 36 contiennent une seule primitive. Les 6 sous-maillages restant correspondent à des combinaisons de plans et cylindres tangents, probablement des congés, qui ne sont alors pas séparés par des arêtes vives.

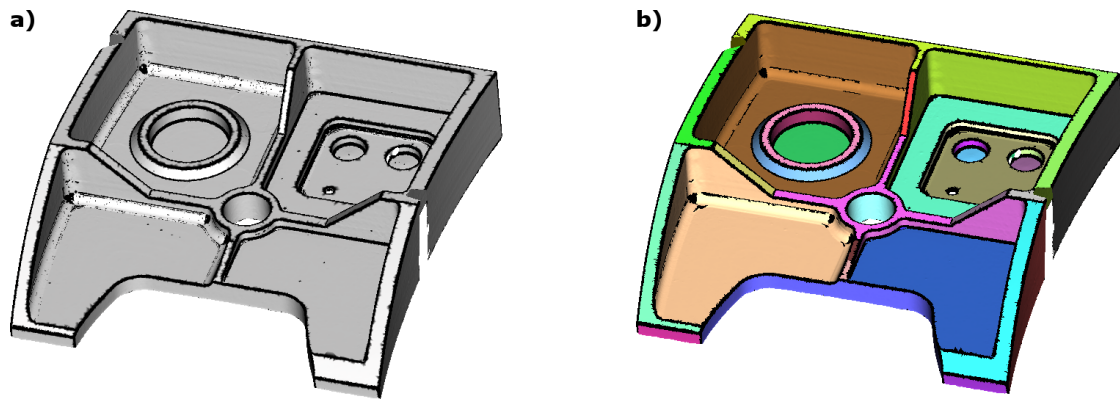


FIGURE 3.17 – Segmentation de *Aerospace* : a) Arêtes vives détectées (en noir), b) Sous-maillages obtenus.

Les arêtes vives de *Moldy* sont correctement détectées (Fig. 3.18.a). Ce maillage étant relativement précis, toutes les arêtes vives sont en effet bien marquées. L'algorithme de segmentation permet alors d'extraire 36 sous-maillages où chacun d'entre eux est le support d'une seule primitive (Fig. 3.18.b). Il semble important de noter que les sous-maillages correspondant aux deux surfaces complexes ne sont pas sur-segmentés. Nous pouvons par contre remarquer que le plan supérieur est coupé au niveau des coins, à cause du processus d'ouverture expliqué dans les sections précédentes. En effet, la fine épaisseur entre les perçages et les bords de la pièce ont été confondus avec des *ponts*. Dans le cadre de la rétro-ingénierie, cela ne pose pas de problème puisque les plans extraits peuvent par la suite être fusionnés.

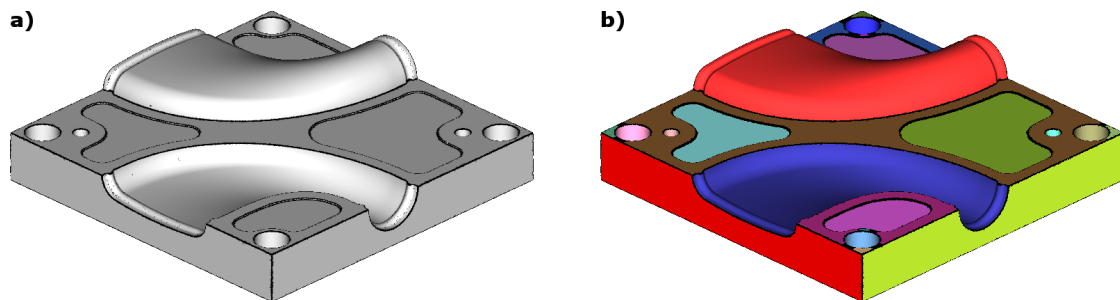


FIGURE 3.18 – Segmentation de *Moldy* : a) Arêtes vives détectées (en noir), b) Sous-maillages obtenus.

Les arêtes vives de *Prise* semblent satisfaisantes, sauf pour les deux cylindres crantés qui sont sur-segmentés (Fig. 3.19.a). D'un autre côté, il semble incohérent de rechercher un cylindre simple à partir d'un cylindre cranté, puisque nous cherchons par la suite à extraire les primitives du modèle original. Cette légère sur-segmentation n'est donc pas un problème dans le cadre de la rétro-ingénierie. Après segmentation, nous obtenons 85 sous-maillages et chacun d'entre eux correspond à une seule primitive (Fig. 3.19.b). Ces résultats sont donc globalement satisfaisants dans le cadre de l'extraction de primitives.

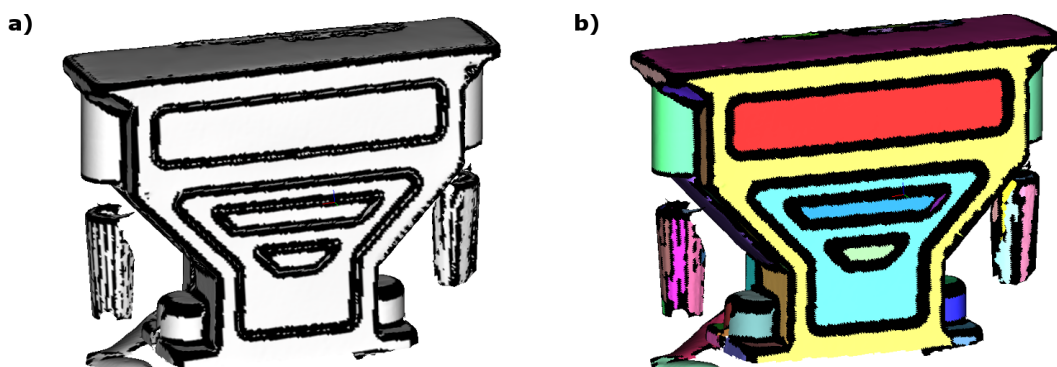


FIGURE 3.19 – Segmentation de *Prise* : a) Arêtes vives détectées (en noir), b) Sous-maillages obtenus.

3.2.4.2 Etude quantitative

Afin de compléter les résultats qualitatifs, nous présentons dans cette section les sous-maillages obtenus après segmentation de trente maillages dont les trois présentés précédemment (Fig. 3.20). Nous pouvons voir que ces maillages se distinguent tant par leur forme que par leur complexité. Ces maillages représentent principalement des objets mécaniques comme un moteur ou un engrenage, mais aussi d'autres objets moins adaptés à la rétro-ingénierie comme une chaise ou une chaussure. De plus, ils ont été générés par plusieurs scanners aux caractéristiques différentes, avec ou sans pré-traitements tels qu'un bouchage de trous ou une décimation par exemple. Enfin, nous les avons volontairement choisis de tailles et résolutions différentes (respectivement $[0, 3; 120]$ cm et $[2.10^4; 2.10^6]$ triangles), mais aussi plus ou moins imprécis (estimation visuelle).

Pourtant, nous pouvons remarquer que ces résultats sont globalement cohérents. En effet, les sous-maillages sont correctement séparés au niveau des arêtes vives des objets, et semblent correspondre pour la plupart à des primitives. L'objectif principal de cette segmentation semble donc atteint, à savoir segmenter automatiquement des maillages de pièces manufacturées en séparant au mieux leurs différents constituants.



FIGURE 3.20 – Résultats de segmentation sur des maillages ayant des caractéristiques différentes.

Afin de valider notre approche, nous avons identifié manuellement les sous-maillages obtenus contenant une seule et unique primitive. Nous pouvons voir dans le tableau 3.2 que, en moyenne, 96% des sous-maillages obtenus correspondent à une seule primitive. Les 4% restants représentent principalement des primitives voisines qui ne sont pas séparées par des arêtes vives, par exemple dans le cas de tangences.

Maillage	1 primitive	Total	en %	Maillage	1 primitive	Total	en %
Fandisk	10	12	83,33	Czslowakei	123	126	97,62
Vase	4	4	100	Chair	143	145	98,62
Lego	33	33	100	Gear	182	185	98,38
Lego_small	14	14	100	Aerospace	36	42	85,71
Cup	14	14	100	Master	83	86	96,51
Manique	15	18	83,33	Moldy	36	36	100
Nespresso	7	7	100	Watertight	185	195	94,87
MediumBolt	5	6	83,33	Aube	55	61	90,16
Hydrant	100	102	98,04	OilPump	463	469	98,72
StripedShoe	20	20	100	Carter	439	440	99,77
Prise	85	85	100	Pump	671	677	99,11
Connector	35	36	97,22	Block	83	90	92,22
SpiceLid	81	82	98,78	Transmission	594	696	85,34
Etui	7	7	100	Splint	23	25	92,00
Blade	118	118	100	Metrologic	835	837	99,76

TABLEAU 3.2 – Identification des sous-maillages contenant une seule primitive.

Des tests de performance ont été effectués sur ces maillages sur un ordinateur standard, avec 8 Go de mémoire RAM et un processeur Intel® Core™ i7-4710 CPU @ 2.50GHz 2.50GHz. Dans le tableau 3.3, nous avons relevé les temps de calcul ainsi que la profondeur maximale de récursion. Nous pouvons voir que la segmentation proposée est rapide, et permet de segmenter des maillages jusqu'à 500k triangles en moins de 10 secondes et jusqu'à 2M triangles en moins d'une minute.

Maillage	Triangles	Temps	Récursion	Maillage	Triangles	Temps	Récursion
Fandisk	19 194	1s	2	Czslowakei	400 026	7s	3
Vase	20 705	1s	1	Chair	500 000	7s	4
Lego	24 371	1s	2	Gear	500 000	9s	3
Lego_small	26 371	1s	2	Aerospace	799 296	10s	1
Cup	55 552	1s	1	Master	820 793	29s	3
Manique	65 090	1s	3	Moldy	851 194	13s	1
Nespresso	71 012	1s	2	Watertight	921 216	19s	4
MediumBolt	89 018	2s	2	Aube	937 176	17s	3
Hydrant	100 000	2s	3	OilPump	1 064 031	27s	4
StripedShoe	100 000	1s	2	Carter	1 067 079	26s	4
Prise	191 021	2s	2	Pump	1 105 570	21s	4
Connector	195 424	3s	2	Block	1 125 832	23s	2
SpiceLid	210 248	3s	2	Transmission	1 500 000	41s	4
Etui	210 963	8s	5	Splint	2 095 079	39s	1
Blade	390 308	6s	1	Metrologic	2 159 724	1m19s	3

TABLEAU 3.3 – Tests de performance de la segmentation.

Théoriquement, la complexité de notre segmentation est de l'ordre de $\mathcal{O}(n \times m)$ avec n le nombre de triangles du maillage et m le nombre de récursions réalisées. Puisqu'un sous-maillage est au moins segmenté en deux parties à chaque récursion, le nombre maximal de récursions est de l'ordre de $\log(n)$. La complexité devient donc $\mathcal{O}(n \log(n))$. Or, nos résultats suggèrent que le nombre de récursions est plutôt constant. En pratique, la complexité de notre algorithme est donc de l'ordre de $\mathcal{O}(n)$. La figure 3.21 montre que le temps de calcul semble effectivement linéaire par rapport au nombre de triangles, avec quelques écarts dûs aux récursions.

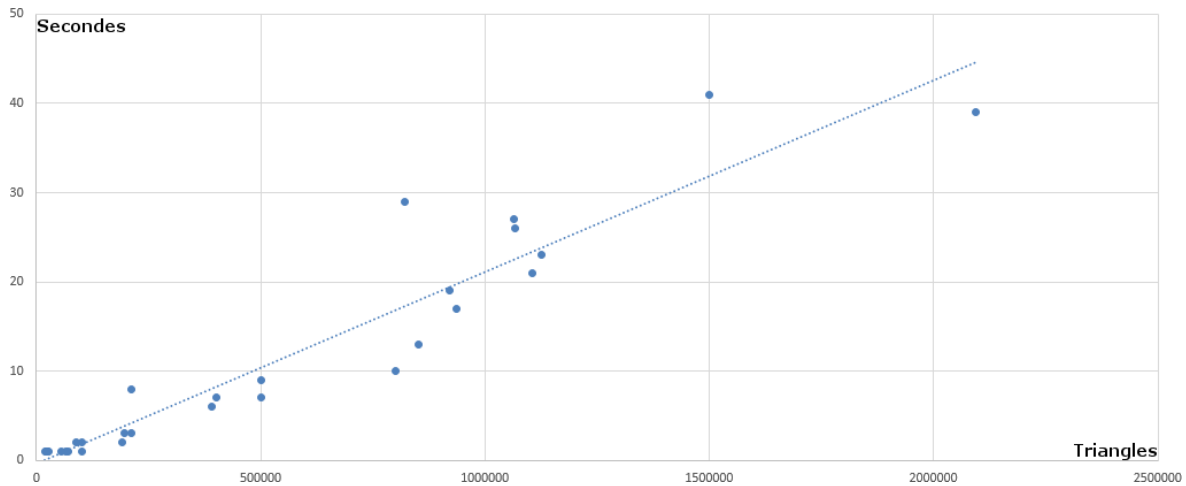


FIGURE 3.21 – Courbe des temps de calcul, en secondes, en fonction du nombre de triangles.

3.3 Ajustement global des primitives

Comme nous l'avons vu en section 3.2.4, la plupart des sous-maillages issus de notre processus de segmentation correspondent à une seule primitive. En partant de ce postulat, nous pouvons directement ajuster une primitive sur chaque sous-maillage. Cet ajustement *global* permet de gagner en précision en prenant en compte des zones de forme homogène les plus larges possibles, mais aussi de réduire les temps de calcul par rapport à une analyse locale.

Dans un premier temps, un ajustement par régression est réalisé sur chaque sous-maillage afin d'estimer les paramètres d'une primitive. Ensuite, ces paramètres sont optimisés afin de minimiser la distance entre les points du maillage et la surface de la primitive. Dans un second temps, nous identifions le type de primitive correspondant au mieux à chaque sous-maillage. Pour cela, nous proposons une analyse des paramètres des primitives en fonction des caractéristiques des sous-maillages.

3.3.1 Optimisation d'une primitive sur un maillage

L'ajustement consiste à déterminer un ensemble de paramètres définissant une primitive, en fonction des points d'un sous-maillage. Notre algorithme commence donc par réaliser des régressions, qui proviennent des travaux de Bénéière *et al.* [5]. Comme expliqué dans la section 1.2.3, ces régressions sont basées sur une combinaison d'approximations par les moindres carrés, permettant d'estimer successivement les paramètres des primitives. Cependant, cette dépendance entre plusieurs paramètres estimés successivement ne permet pas toujours de garantir leur fiabilité. La figure 3.22.a montre par exemple un cône dont les paramètres ont été estimés avec la méthode de Bénéière *et al.* [5]. Or, les paramètres de position définissant le sommet du cône dépendent des paramètres définissant l'axe, qui sont faux dans cet exemple. Dans ce cas, l'ajustement par régression n'est pas satisfaisant.

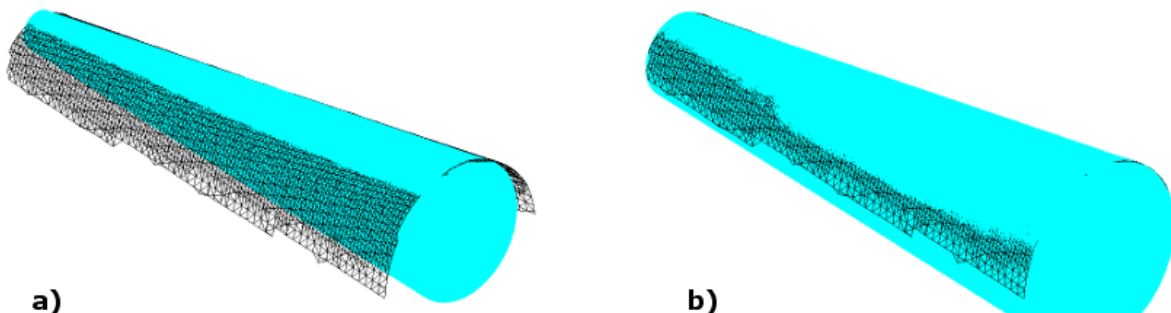


FIGURE 3.22 – Ajustement d'un cône sur un maillage : a) Par régressions successives, b) Par optimisation.

D'un autre côté, les méthodes d'optimisation nécessitent généralement une bonne initialisation des paramètres des primitives pour pouvoir converger. Dans ce sens, nous proposons d'ajuster les primitives d'abord par une combinaison de régressions, puis par optimisation. La figure 3.22.b montre le cône obtenu après optimisation, dont les paramètres sont mieux estimés.

Cette optimisation consiste à déterminer les paramètres qui minimisent la somme quadratique des distances entre les points P_i du maillage et la surface S_j de la primitive :

$$S'_j = \operatorname{argmin}_{\{p_1 \dots p_k\}} \left\{ \sum_{i=1}^n \mathcal{D}_j(S_j, P_i)^2 \right\} \quad \text{avec} \quad S_j = \{p_1 \dots p_k\}, \quad (3.5)$$

avec j le type de primitive, n le nombre de points du sous-maillage, \mathcal{D} la fonction de distance associée aux paramètres $\{p_1 \dots p_k\}$ de S_j . Les fonctions de distances utilisées ici proviennent de la méthode de Shakarji [96]. Notons en particulier que ces distances doivent être signées, en fonction de l'intérieur ou de l'extérieur des primitives, pour que l'optimisation puisse converger.

Dans le cas du plan, la fonction de distance \mathcal{D} dépend de sa normale \vec{n} et d'un point P appartenant au plan :

$$\mathcal{D}_{plan}(\vec{n}, P, P_i) = \frac{\vec{n} \cdot \overrightarrow{PP_i}}{\|\vec{n}\|} \quad \text{avec} \quad S_{plan} = \{\vec{n}, P\}. \quad (3.6)$$

Dans le cas de la sphère, la fonction de distance \mathcal{D} dépend de son rayon r et de son centre P :

$$\mathcal{D}_{sphère}(r, P, P_i) = \|\overrightarrow{PP_i}\| - r \quad \text{avec} \quad S_{sphère} = \{r, P\}. \quad (3.7)$$

Dans le cas du cylindre, la fonction de distance \mathcal{D} dépend de son rayon r et de son axe, défini par une direction \vec{n} et un point P :

$$\mathcal{D}_{cylindre}(r, \vec{n}, P, P_i) = \frac{\|\overrightarrow{PP_i} \wedge \vec{n}\|}{\|\vec{n}\|} - r \quad \text{avec} \quad S_{cylindre} = \{r, \vec{n}, P\}. \quad (3.8)$$

Dans le cas du cône, la fonction de distance \mathcal{D} dépend de son semi-angle ψ et de son axe, défini par une direction \vec{n} et le sommet du cône. D'après l'article de Shakarji [96], l'optimisation ne peut pas converger en utilisant le sommet directement. Pour ajouter un degré de liberté à la fonction de distance, le sommet est alors remplacé par un point P appartenant à l'axe et la distance d entre P et le sommet du cône :

$$\mathcal{D}_{cône}(d, \psi, \vec{n}, P, P_i) = \frac{\|\overrightarrow{PP_i} \wedge \vec{n}\|}{\|\vec{n}\|} \cos(\psi) + \frac{\vec{n} \cdot \overrightarrow{PP_i}}{\|\vec{n}\|} \sin(\psi) - d \quad \text{avec} \quad S_{cône} = \{d, \psi, \vec{n}, P\}. \quad (3.9)$$

Ce problème de minimisation peut être résolu par des méthodes classiques tels qu'une descente de gradient [99] ou encore l'algorithme de Levenberg-Marquardt [76]. Le principe général est constitué de trois étapes. Tout d'abord, la somme quadratique des distances est calculée. Ensuite, le vecteur gradient de la fonction de distance est calculé à l'aide de calculs matriciels. Enfin, les paramètres de la primitive sont modifiés dans la direction du vecteur gradient pour minimiser les distances. Ces trois étapes sont alors réitérées, jusqu'à convergence du système vers une solution stable. Pour cela, les distances calculées lors de deux itérations successives sont comparées et l'algorithme s'arrête si leur différence est inférieure à un seuil prédéfini. Ainsi, plus les paramètres initiaux sont précis et plus le système converge rapidement. D'un autre côté, le système peut ne pas converger si les paramètres initiaux sont trop éloignés de la solution. Dans notre cas, le système converge toujours et seules quelques itérations sont nécessaires puisque les paramètres fournis par les régressions sont relativement proches des paramètres optimaux.

Nous avons ainsi ajusté des primitives sur les sous-maillages issus de la segmentation des 30 objets présentés dans la section 3.2.4. Ensuite, nous avons calculé la distance entre chaque point de chaque sous-maillage et la surface de la primitive correspondante. Avec un ajustement par régression puis optimisation, nous obtenons une distance moyenne inférieure d'un facteur 2 par rapport à un ajustement par régression uniquement. L'optimisation semble donc améliorer la précision des primitives par rapport à la régression seule. En effet, tous les paramètres sont estimés en même temps, ce qui permet d'éviter une éventuelle propagation d'erreur.

3.3.2 Identification du type de primitive

Lors de cet ajustement global, nous ne connaissons pas *a priori* le type de primitive correspondant à chaque sous-maillage. Pour l'identifier, nous proposons d'ajuster globalement chaque type de primitive puis d'en déterminer le plus adéquat (Algorithme 5).

Algorithme 5 Identification du type de primitive

```

Soit un ensemble de sous-maillages  $\mathcal{M}$ 
pour tout sous-maillage  $M_i \in \mathcal{M}$  faire
     $\delta_i \leftarrow \infty$ 
    pour tout primitive  $S_j \in \{S_{plan}, S_{sphère}, S_{cylindre}, S_{cône}\}$  ajustée sur  $M_i$  faire
         $\delta_j \leftarrow \overline{\mathcal{D}_j}(M_i, S'_j)$ 
        si  $\delta_j \leq \delta_i$  et  $\delta_j \leq \Delta_{distance}$  alors
            Associer  $S'_j$  à  $M_i$ 
             $\delta_i \leftarrow \delta_j$ 
        fin si
    fin pour
fin pour
    
```

Cela implique, pour chaque sous-maillage, d'ajuster un plan, une sphère, un cylindre et un cône. Ensuite, une mesure de déviation est calculée entre le sous-maillage et chaque primitive. Ces quatre mesures sont alors comparées entre elles pour déterminer la primitive correspondant au mieux au sous-maillage. Pour calculer la déviation d'une primitive par rapport à un sous-maillage, la distance moyenne $\overline{\mathcal{D}_j}(M_i, S'_j)$ entre les points du maillage M_i et la surface de la primitive S'_j est calculée (Fig. 3.23.a), et doit être inférieure à $\Delta_{distance}$.

Pendant, le calcul de distance seul n'est pas suffisant pour garantir une bonne identification du type de primitive, puisque l'optimisation permet de réduire cette distance pour chaque type. Nous proposons alors deux mesures supplémentaires, basées sur l'analyse des normales du maillage. Ainsi, la déviation angulaire est calculée entre la normale de chaque point sur le maillage et la normale correspondante théorique sur la surface de la primitive (Fig. 3.23.b). La moyenne de ces déviations doit alors être inférieure à Δ_{angle} . Enfin, une troisième vérification est réalisée en comparant l'orientation des primitives obtenues par régression et celle après optimisation (Fig. 3.23.c). L'angle entre ces deux orientations doit alors également être inférieure à Δ_{angle} .

Si une de ces trois mesures a une valeur trop importante, selon la tolérance correspondante, alors le type est supposé inconnu et aucune primitive n'est conservée.

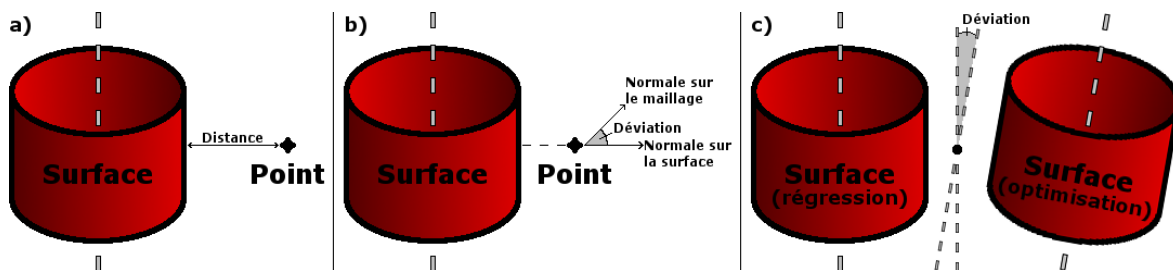


FIGURE 3.23 – Analyse de déviation en trois étapes : a) Distance, b) Déviation angulaire, c) Déviation axiale.

En théorie, une zone plane peut être approximée par un cylindre ou une sphère de rayon infini, posant un problème d'identification. Certains sous-maillages peuvent alors être approximés par des primitives de très grande taille, qui ne sont pas présentes dans le modèle original (Fig. 3.24). Pour compenser ce problème, une dernière vérification consiste à rejeter une primitive si l'aire du sous-maillage ne représente qu'une infime partie de l'aire de la surface de la primitive. Prenons par exemple le cas d'un sous-maillage approximé par une sphère de rayon r . L'aire totale de la sphère est égale à $4\pi r^2$. Si l'aire du sous-maillage est inférieure à un certain pourcentage de cette aire, alors la primitive n'est pas conservée. Dans le cas d'un cylindre ou d'un cône, nous délimitons au préalable la surface de la primitive à l'aide de la boîte englobante du sous-maillage.

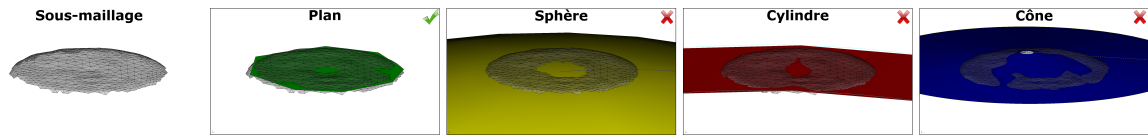


FIGURE 3.24 – Approximation incohérente d'un sous-maillage par des primitives de très grande taille.

Afin de valider notre approche, nous avons réalisé un ajustement global avec identification automatique du type de primitive sur les sous-maillages issus de la segmentation (voir section 3.2.4). Nous avons ensuite déterminé manuellement un taux d'identification correcte. Rappelons que 96% de ces sous-maillages, en moyenne, ne contiennent qu'une seule primitive. Les 4% restants doivent donc être identifiés comme un type inconnu.

La figure 3.25 montre trois objets dont les sous-maillages sont analysés. Les résultats obtenus suggèrent une identification fiable du type de primitive. Nous remarquons en particulier que les surfaces à forme libre de *Moldy* ne sont pas confondues avec des primitives simples (Fig. 3.25.b). De même, un sous-maillage de *Metrologic* contient un tronçon de tore et deux sphères (Fig. 3.25.c). L'ajustement global échoue alors à estimer une primitive unique, et le sous-maillage est identifié de type inconnu. Pour ces trois objets, nous obtenons un taux d'identification de 100%.

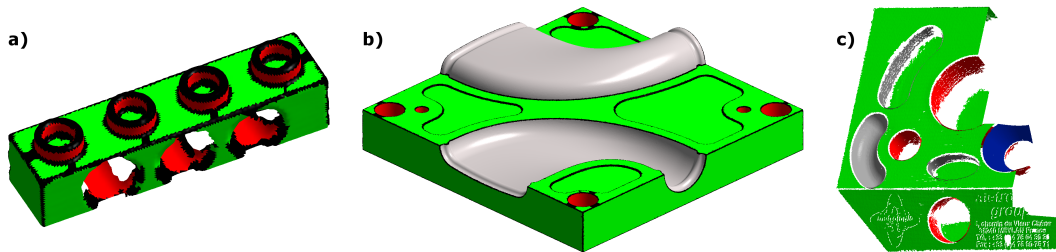


FIGURE 3.25 – Identification du type de primitive (vert pour plan, rouge pour cylindre, bleu pour cône et gris pour inconnu), pour chaque sous-maillage de : a) *Lego*, b) *Moldy*, c) *Metrologic*.

Afin de mesurer l'impact de chacune des quatre vérifications, nous avons mesuré le taux d'identification pour chaque objet en les ajoutant progressivement. Les résultats obtenus, résumés dans le tableau 3.4, montrent que chaque vérification permet d'améliorer l'identification. Au final, notre méthode obtient un taux d'identification moyen de 98,9%. Les sous-maillages mal identifiés représentent principalement des zones très bruitées qui ne sont pas censées être supports de primitives.

Maillage	Identification (en %)				Maillage	Identification (en %)			
	(a)	(b)	(c)	(d)		(a)	(b)	(c)	(d)
Fandisk	100	100	100	100	Czslowakei	96,6	96,6	97,1	100
Vase	100	100	100	100	Chair	98,2	100	100	100
Lego	88,0	95,8	100	100	Gear	93,9	98,1	100	100
Lego_small	100	100	100	100	Aerospace	100	100	100	100
Cup	81,8	81,8	98,4	98,4	Master	72,2	86,3	88,7	88,8
Manique	93,7	93,7	100	100	Moldy	96,3	100	100	100
Nespresso	80,0	80,0	100	100	Watertight	85,2	88,6	91,8	98,0
MediumBolt	100	100	100	100	Aube	100	100	100	100
Hydrant	82,2	85,0	92,3	97,1	OilPump	83,5	87,1	89,8	91,2
StripedShoe	100	100	100	100	Carter	85,9	95,1	95,7	96,4
Prise	83,1	83,1	100	100	Pump	97,3	97,8	98,5	98,7
Connector	86,7	86,7	89,3	100	Block	100	100	100	100
Spicelid	76,8	93,0	93,1	97,9	Transmission	97,8	98,6	99,1	99,3
Etui	100	100	100	100	Splint	94,1	94,1	98,6	100
Blade	96,5	96,5	98,2	100	Metrologic	100	100	100	100

TABLEAU 3.4 – Taux d'identification correcte des types de primitives en fonction des vérifications réalisées : a) Calcul de distance, b) Ajout de l'analyse des normales, c) Ajout de l'analyse des orientations des primitives, d) Ajout de l'analyse des aires.

3.4 Segmentation “arêtes lisses”

Nous avons vu dans la section 3.2.4 que notre segmentation fournit certains sous-maillages qui contiennent plusieurs primitives, séparées par des arêtes lisses (environ 4% des cas). Dans ce cas, l'ajustement global ne permet pas d'identifier correctement un type de primitive, et aucune primitive n'est extraite. Nous cherchons alors à segmenter à nouveau ces sous-maillages, à l'aide d'un second algorithme (Algorithme 6).

Algorithme 6 Segmentation par seuillage multiple

```

Soit un ensemble de sous-maillages  $\mathcal{M}$ 
pour tout sous-maillage  $M_i \in \mathcal{M}$  faire
  Construire et analyser la distribution de courbure moyenne de  $M_i$ 
  pour tout paire de vallées successives  $(V_i, V_{i+1})$  faire
     $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup$  Segmentation de  $M_i$  en fonction de deux seuils définis par  $V_i$  et  $V_{i+1}$ 
  fin pour
   $\mathcal{M}_i \leftarrow$  Suppression des chevauchements entre les sous-maillages de  $\mathcal{M}_i$ 
  si  $\mathcal{M}_i \neq \emptyset$  alors
     $\mathcal{M} \leftarrow \mathcal{M}_i \cup \mathcal{M} \setminus M_i$ 
  fin si
fin pour
renvoyer  $\mathcal{M}$ 

```

Cette seconde segmentation est basée sur la même analyse des courbures que la première, présentée dans la section 3.2.1. Nous calculons ainsi un ensemble de vallées $\{V_1 \dots V_k\}$, à partir de l'histogramme de courbure moyenne normalisée. Lors de la segmentation “arêtes vives”, deux seuils sont définis à partir des vallées extrêmes (V_1, V_k) (voir section 3.2.2). Pour détecter les “arêtes lisses”, nous pourrions définir une catégorisation des points du maillage en définissant un identifiant pour chaque intervalle de courbure moyenne normalisée. Cependant, la catégorisation d'un triangle est réalisée en recherchant la catégorie majoritaire de ses points. Or, s'il existe plus de deux catégories, ce qui serait le cas ici, il est impossible de déterminer avec certitude la bonne catégorie. Dans ce cas, il semble préférable de réaliser des seuillages successifs.

Nous proposons alors de fixer des seuils à partir de deux vallées (V_i, V_{i+1}) , qui définissent un intervalle de courbure moyenne normalisée homogène. Le processus est ainsi itéré pour chaque paire de vallées (V_i, V_{i+1}) . Nous obtenons alors un ensemble de seuillages du maillage, qui permet de construire des sous-maillages pour chaque intervalle. La figure 3.26 montre un exemple de segmentation par seuillage multiple, sur un maillage contenant deux plans et un cylindre.

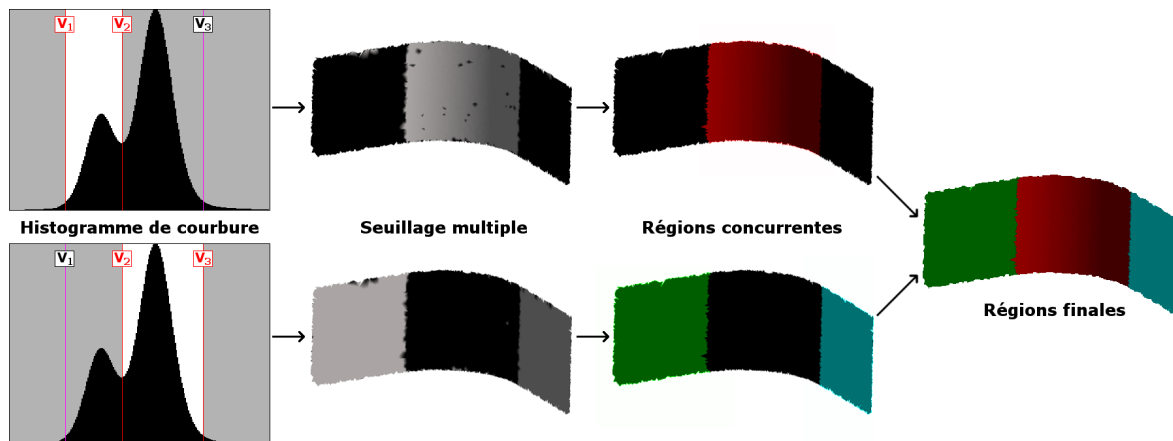


FIGURE 3.26 – Reconstruction des régions uniformes par seuillage multiple.

Nous pouvons voir que l'histogramme de courbure moyenne normalisée contient deux modes, correspondant respectivement aux courbures cylindriques et planes. Dans la première étape, les seuils de segmentation sont fixés en fonction des vallées V_1 et V_2 . Le processus de seuillage définit alors une seule région uniforme, cylindrique, entourée de deux régions arêtes. Dans la seconde étape, les seuils sont fixés selon les vallées V_2 et V_3 . Deux régions planes sont ainsi extraites, séparées par une région arête. Au final, chaque triangle est assigné à la région uniforme correspondante, ce qui permet de reconstruire trois sous-maillages, correspondant chacun à une seule primitive. Les "arêtes lisses" correspondent alors à la séparation entre deux sous-maillages adjacents.

Cependant, un triangle peut parfois faire partie de plusieurs régions lors du seuillage multiple à cause de nos post-traitements (voir section 3.2.2). Nous devons alors déterminer la région la plus pertinente pour chaque triangle. Nous proposons pour cela de calculer un score pour chaque région, qui correspond au degré d'érosion nécessaire pour l'éliminer totalement (Fig. 3.27). Ce score représente en quelque sorte "l'épaisseur maximale" d'une région, en nombre de triangles. Chaque triangle peut donc obtenir plusieurs scores, et il est finalement assigné à la région correspondant au meilleur score.

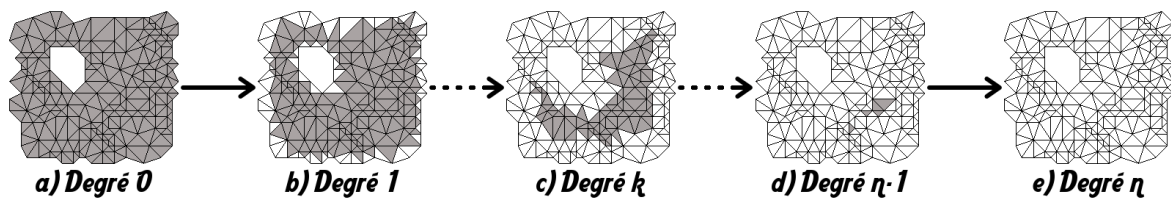


FIGURE 3.27 – Calcul du score d'une région correspondant au degré d'érosion maximal.

Comme pour la première segmentation, nous avons identifié les sous-maillages correspondant à une seule primitive après la seconde segmentation. Le tableau 3.5 compare les résultats obtenus entre les deux étapes de segmentation. Après la première segmentation, nous identifions en moyenne 95,8% de sous-maillages correspondant à une seule primitive. Après la seconde segmentation réalisée sur les 4,2% restants, nous obtenons un score total d'identification de 99,7% en moyenne. Chacun de ces sous-maillages peut donc être correctement approximé par une primitive unique à l'aide de l'ajustement global présenté dans la section 3.3.

Maillage	Après étape 1 (en %)	Après étape 2 (en %)	Maillage	Après étape 1 (en %)	Après étape 2 (en %)
Fandisk	83,33	100	Czslowakei	97,62	100
Vase	100	100	Chair	98,62	98,73
Lego	100	100	Gear	98,38	100
Lego_small	100	100	Aerospace	85,71	98,95
Cup	100	100	Master	96,51	99,80
Manique	83,33	100	Moldy	100	100
Nespresso	100	100	Watertight	94,87	98,12
MediumBolt	83,33	100	Aube	90,16	100
Hydrant	98,04	100	OilPump	98,72	100
StripedShoe	100	100	Carter	99,77	99,77
Prise	100	100	Pump	99,11	99,72
Connector	97,22	100	Block	92,22	100
SpiceLid	98,78	100	Transmission	85,34	98,43
Etui	100	100	Splint	92,00	97,56
Blade	100	100	Metrologic	99,76	99,76

TABLEAU 3.5 – Sous-maillages contenant une seule primitive à chaque étape de segmentation.

3.5 Adaptation automatique de l'ajustement local des primitives

Après les segmentations du maillage, par arêtes vives et lisses (sections 3.2 et 3.4), et l'ajustement global des primitives sur les sous-maillages obtenus (section 3.3), il peut rester des sous-maillages de type inconnu qui ne sont pas associés à une primitive (environ 0,3%). Dans ce cas, ces sous-maillages de type inconnu peuvent définir :

- soit une surface complexe non représentable par des primitives;
- soit une combinaison de primitives non dissociées.

Dans cette section, nous cherchons à traiter le deuxième cas. Puisque la méthode d'ajustement global ne fonctionne pas, nous réalisons une analyse locale de chaque sous-maillage de type inconnu.

3.5.1 Méthode d'ajustement local de primitives à partir d'un sous-maillage

Pour cela, nous reprenons la méthode d'extraction des primitives géométriques de Bénéière *et al.* [5] qui procède par croissance de régions sur les points d'un sous-maillage, en fonction de leurs valeurs de courbures principales. Cette méthode se décompose en huit étapes :

1. calculer les courbures principales (minimale et maximale) en chaque point;
2. choisir un point au hasard et définir en ce point le type de primitive selon ses courbures;
3. regrouper les points voisins ayant des caractéristiques de courbures similaires;
4. estimer les paramètres d'une primitive du type prédéfini à partir d'un ensemble de points;
5. continuer à regrouper les points voisins s'ils correspondent à la primitive estimée;
6. mettre à jour les paramètres de la primitive en fonction de ces nouveaux points ajoutés;
7. itérer les étapes 5 et 6 jusqu'à ne plus pouvoir ajouter de nouveau point;
8. itérer les étapes 2 à 7 avec un nouveau point qui n'a pas encore été traité.

Plusieurs étapes sont paramétrées par des tolérances. Lors de l'étape 2, pour déterminer le type de primitive en fonction des courbures principales, Bénéière *et al.* définissent un écart maximal entre deux valeurs de courbure similaires. De même, pour l'étape 5, l'appartenance d'un point à une primitive est conditionnée à la distance qui les sépare. Bénéière *et al.* définissent donc les tolérances de courbure et de distance ci-dessous pour les différents types de primitive :

1. Plans :
 - $\mathcal{C}_{zero}^{plan}$: valeur maximale de courbure considérée comme nulle;
 - $\mathcal{D}_{zero}^{plan}$: distance maximale entre un point et le plan associé.
2. Sphères :
 - $\mathcal{C}_{proche}^{sphère}$: écart relatif maximal entre deux valeurs de courbures similaires;
 - $\mathcal{D}_{zero}^{sphère}$: distance maximale entre un point et la sphère associée.
3. Cônes (un cylindre est considéré comme un cône d'angle nul) :
 - $\mathcal{C}_{zero}^{cône}$: valeur maximale de courbure considérée comme nulle;
 - $\mathcal{D}_{zero}^{cône}$: distance maximale entre un point et le cône associé;
 - $\mathcal{A}_{proche}^{cône}$: écart maximal entre deux angles similaires.

Ainsi, un point P_i choisi au hasard, associé à deux valeurs de courbures principales k_1 et k_2 , est déterminé comme appartenant à un plan si $k_1 \leq \mathcal{C}_{zero}^{plan}$ et $k_2 \leq \mathcal{C}_{zero}^{plan}$. Ensuite, si des points voisins $\{P_i \dots P_j\}$ sont tous déterminés comme appartenant à un plan, alors ils sont agrégés pour former une région sur laquelle un plan \mathcal{P} est ajusté. Une fois le plan \mathcal{P} ajusté, un nouveau point P peut être agrégé à la région si la distance entre P et \mathcal{P} est inférieure à $\mathcal{D}_{zero}^{plan}$.

Cependant, dans la méthode de Bénéière *et al.* [5], ces tolérances sont définies globalement. De plus, ces valeurs sont estimées pour des maillages issus de discrétisation CAO. Dans notre cas, nous avons des sous-maillages qui sont imprécis. Nous devons donc adapter la méthode d'ajustement local de primitives pour traiter efficacement ces sous-maillages.

Pour cela, nous ajoutons deux étapes supplémentaires à la méthode de Bènière *et al.*, entre les étapes 1 et 2 :

- 1.1 identification du type de primitive majoritaire pour chaque sous-maillage de type inconnu ;
- 1.2 calcul automatique des tolérances en fonction du type majoritaire identifié.

Nous définissons également de nouvelles tolérances, plus générales et définies indépendamment pour chaque sous-maillage de type inconnu :

- \mathcal{C}_{zero} : valeur absolue maximale de courbure considérée comme nulle ;
- \mathcal{C}_{proche} : écart maximal entre deux valeurs de courbures similaires ;
- \mathcal{D}_{zero} : distance maximale acceptable entre un point et la primitive associée ;
- \mathcal{A}_{proche} : écart maximal entre deux angles similaires.

Dans la suite de cette section, nous décrivons les deux étapes proposées.

3.5.2 Identification du type de primitive majoritaire

Comme nous avons vu dans la section 1.2.2, les distributions des courbures principales permettent d'identifier le type de primitive associé à un ensemble de points de manière robuste. Nous proposons donc de construire les distributions des courbures minimale et maximale, pour chaque sous-maillage considéré, grâce à une estimation par noyau comme nous l'avons fait pour la distribution de courbure moyenne dans le cas de la segmentation (voir section 3.2.1). Ensuite, nous pouvons chercher le mode pour chaque distribution minimale et maximale, et nous cherchons à identifier le type de primitive en fonction de la position de ces deux modes.

Nous pouvons alors distinguer les quatre cas suivants, qui permettent de définir le type de primitive majoritaire :

- cas du plan : les deux modes sont proches de zéro ;
- cas de la sphère : les deux modes sont proches mais différents de zéro ;
- cas du cylindre / cône : les deux modes sont éloignés et l'un d'eux est proche de zéro ;
- autres cas : les deux modes sont éloignés et aucun n'est proche de zéro.

Le dernier cas ne correspond à aucun type de primitive connu. Pour autant, lorsqu'un sous-maillage contient plusieurs primitives, l'analyse du type majoritaire peut être ambiguë. En effet, la détection d'un seul mode ne garantit pas la présence d'un seul type de primitive, notamment à cause des chevauchements dûs aux imprécisions (voir section 3.2.1). C'est pour cela que nous parlons ici de type "majoritaire", représentant la majeure partie du sous-maillage mais pas sa totalité.

Dans le cas du cône, une des courbures principales est nulle et l'autre est variable. Dans la figure 3.28.a, nous pouvons en effet voir qu'une discrétisation régulière d'un cône engendre plusieurs modes dans la distribution de courbure maximale. Dans ce cas, chaque mode correspond à un cercle du cône, et leur importance dépend de la quantité de points sur chaque cercle. Dans cet exemple, chaque cercle est représenté avec le même nombre de points, qui sont exactement alignés le long de la surface du cône. Les courbures minimales des points sont alors nulles et la distribution associée correspond à un mode unique et centré en zéro.

En pratique, les scanners fournissent généralement des points répartis sur la surface de manière relativement uniforme. Dans ce cas, nous obtenons plus de points sur la partie évasée du cône, impliquant alors des distributions de courbures similaires à celles d'un cylindre (Fig. 3.28.b). Il est à noter que puisque ces points ne sont pas parfaitement alignés le long de la surface du cône, leurs valeurs de courbure minimale ne sont pas exactement nulles. La distribution correspondante peut alors s'étaler légèrement et ne pas être centrée exactement en zéro.

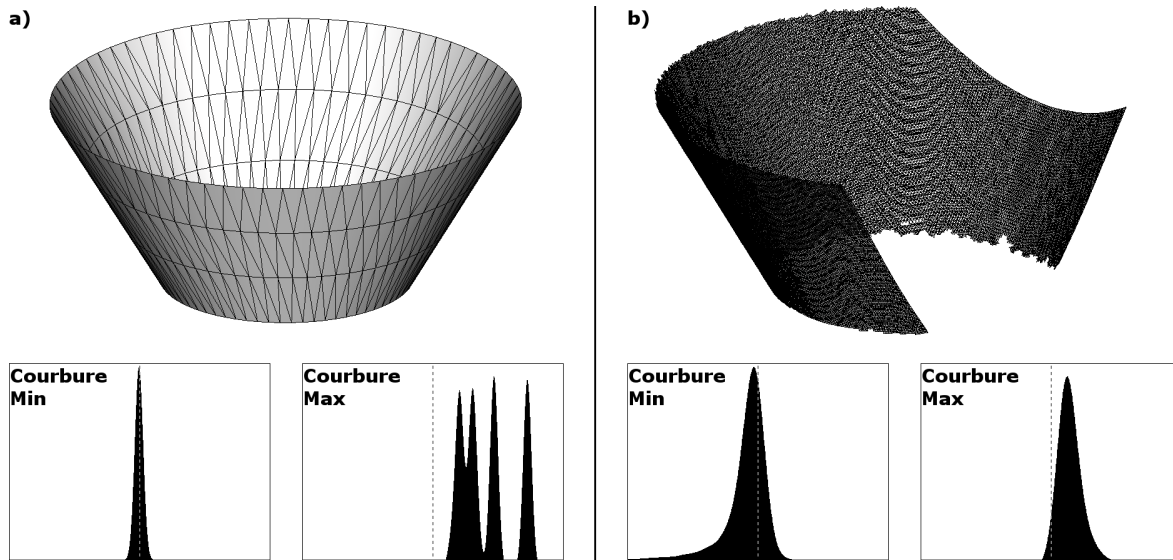


FIGURE 3.28 – Distributions des courbures principales d'un cône : a) Maillage CAO, b) Maillage numérisé.

3.5.3 Calcul automatique des tolérances

Afin de rendre l'extraction de primitives automatique et adaptative, nous proposons de calculer les tolérances indépendamment pour chaque sous-maillage considéré, à partir du type de primitive identifié et de statistiques calculées à partir de la distribution des courbures principales. Nous supposons ici qu'une distribution de courbure est généralement définie par une Gaussienne (voir section 1.2.2). Dans un premier temps, nous définissons les symboles suivants :

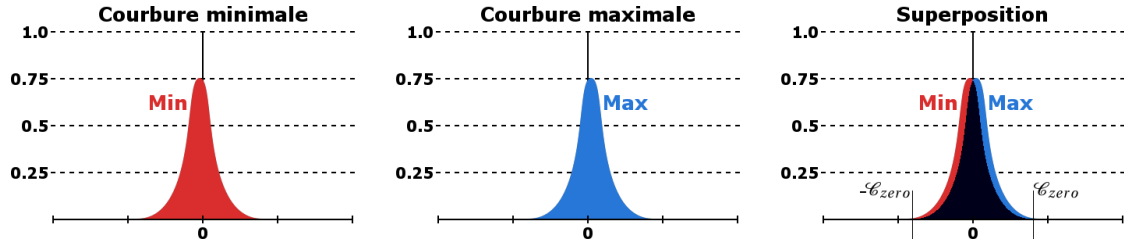
- μ_{min} et μ_{max} : moyennes des courbures minimale et maximale ;
- σ_{min} et σ_{max} : écarts-types des courbures minimale et maximale.

Afin de calculer des tolérances plus robustes et ainsi permettre l'extraction de primitives de différents types, nous basons nos calculs sur l'intervalle de courbure autour du mode de chaque distribution, défini par une paire de vallées (voir section 3.2.1). La figure 3.29 montre deux estimations des paramètres d'une Gaussienne, à partir de la distribution de courbure maximale d'un sous-maillage numérisé. Ainsi, nous pouvons voir que les paramètres de la Gaussienne obtenus à partir de l'intervalle de courbure (Fig. 3.29.b) sont plus robustes que ceux obtenus à partir de l'ensemble des valeurs de courbure (Fig. 3.29.a).



FIGURE 3.29 – Estimation des paramètres d'une Gaussienne pour un sous-maillage numérisé : a) A partir de toutes les valeurs de courbure, b) A partir de l'intervalle de courbure correspondant au mode.

Dans le cas du type majoritairement plan, les courbures minimale et maximale sont proches de zéro. Nous pouvons alors calculer les deux tolérances de courbure \mathcal{C}_{zero} , qui définit le seuil en dessous duquel une valeur de courbure est considérée nulle, et \mathcal{C}_{proche} , qui définit l'écart maximal entre deux valeurs de courbure similaires. La figure 3.30 illustre la détermination de \mathcal{C}_{zero} par une superposition des distributions des courbures principales.

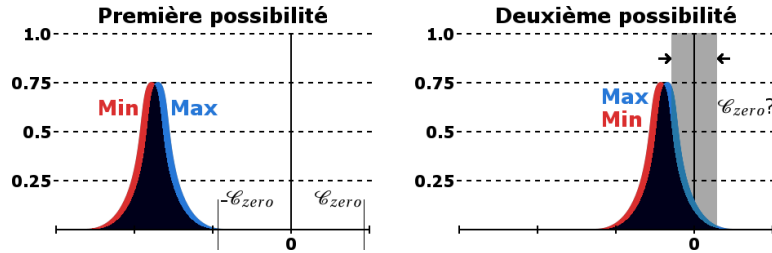

 FIGURE 3.30 – Signification de la tolérance \mathcal{C}_{zero} , pour un plan, par superposition des distributions.

Le calcul de ces deux tolérances est donné par :

$$\mathcal{C}_{zero} = \frac{\max(\mu_{min} + 3\sigma_{min}; \mu_{max} + 3\sigma_{max}) - \min(\mu_{min} - 3\sigma_{min}; \mu_{max} - 3\sigma_{max})}{2}, \quad (3.10)$$

$$\mathcal{C}_{proche} = \min(\sigma_{min}; \sigma_{max}).$$

Dans le cas du type majoritairement sphérique, les courbures principales sont similaires et différentes de zéro. Pour les maillages imprécis, nous obtenons approximativement deux distributions dont les moyennes sont très proches. Cependant, une sphère avec un rayon élevé peut engendrer une distribution de courbures très proche de zéro. Dans ce cas, il est nécessaire de prendre en compte le chevauchement entre la Gaussienne obtenue et l'abscisse zéro (Fig. 3.31).


 FIGURE 3.31 – Calcul de la tolérance \mathcal{C}_{zero} pour une sphère.

Nous proposons alors de calculer les tolérances par :

$$\mathcal{C}_{zero} = \min(|\mu_{min} \pm 3\sigma_{min}|; |\mu_{max} \pm 3\sigma_{max}|; \sigma_{min}; \sigma_{max}), \quad (3.11)$$

$$\mathcal{C}_{proche} = \min(\sigma_{min}; \sigma_{max}).$$

Dans le cas du type majoritairement cylindrique (ou cônique), nous obtenons une distribution proche de zéro, définie par une Gaussienne $(\mu_{zero}, \sigma_{zero})$, et une autre plus éloignée, définie par $(\mu_{nonzero}, \sigma_{nonzero})$. Nous devons alors déterminer des tolérances permettant de séparer au mieux ces deux distributions, pouvant se chevaucher mutuellement (Fig. 3.32).

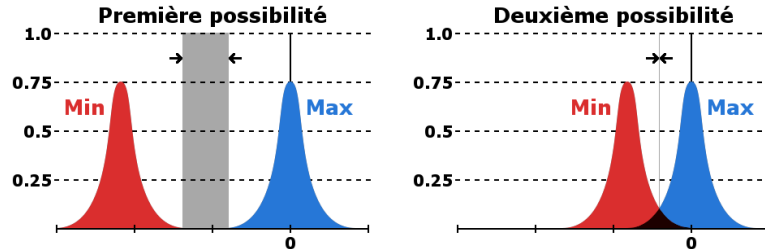


FIGURE 3.32 – Zone de séparation optimale entre deux Gaussiennes, avec ou sans chevauchement.

Nous proposons alors de déterminer ces tolérances avec une moyenne des positions des deux modes, pondérée par leurs écarts-types :

$$\mathcal{C}_{zero} = \min(\mu_{zero} + 3\sigma_{zero}; \frac{|\mu_{min}| \times \sigma_{max} + |\mu_{max}| \times \sigma_{min}}{\sigma_{min} + \sigma_{max}}), \quad (3.12)$$

$$\mathcal{C}_{proche} = \sigma_{nonzero}.$$

Pour la tolérance \mathcal{D}_{zero} , qui définit la distance maximale entre un point et la primitive associée à celui-ci, nous ne pouvons pas nous baser directement sur une analyse des valeurs de courbure. Nous proposons alors d'adapter cette tolérance de distance à partir de la longueur moyenne \mathcal{L} des arêtes du sous-maillage de type inconnu, utilisée précédemment pour la normalisation des distributions de courbures (voir section 3.2.1) :

$$\mathcal{D}_{zero} = \mathcal{L}. \quad (3.13)$$

Pour la tolérance \mathcal{A}_{proche} , qui définit l'écart maximal entre deux angles égaux, nous proposons d'analyser les directions des courbures principales en chaque point :

$$\mathcal{A}_{proche} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\alpha_i - \bar{\alpha})^2} \quad \text{avec} \quad \bar{\alpha} = \frac{1}{n} \sum_{i=1}^n \alpha_i, \quad (3.14)$$

avec n le nombre de points du sous-maillage de type inconnu et α_i l'angle minimal entre les directions de courbure du $i^{ème}$ point et celles d'un de ses voisins. Nous calculons donc, pour chaque paire de points voisins (P_i, P_j) , l'angle minimal parmi les quatre possibles entre leurs directions de courbure $(\vec{D}_{1min}, \vec{D}_{1max})$ et $(\vec{D}_{2min}, \vec{D}_{2max})$. La tolérance \mathcal{A}_{proche} correspond alors à l'écart-type de ces angles.

3.5.4 Exemple d'application

Afin de valider notre méthode, nous avons analysé des sous-maillages de type inconnu. Après la segmentation de *Metrologic*, par arêtes vives et lisses, et l'ajustement global de primitives, il reste trois sous-maillages de type inconnu (Fig. 3.33.a). En analysant ces trois sous-maillages, nous identifions leur type majoritaire et calculons des tolérances. Les résultats obtenus suggèrent une identification fiable du type de primitive. Nous remarquons en particulier un sous-maillage identifié comme majoritairement sphérique, qui comprend effectivement deux sphères. Ce dernier contient également une portion de tore, mais qui est minoritaire lors de l'analyse des courbures. L'identification permet dans ce cas d'adapter les tolérances puis d'extraire correctement les deux sphères, en laissant une portion de maillage sans primitive (Fig. 3.33.b). Nous pouvons également remarquer que les tolérances sont calculées sur chaque sous-maillage indépendamment. Par exemple, l'écart-type de la courbure maximum du sous-maillage majoritairement cylindrique à droite est approximativement égal au double de celui du sous-maillage majoritairement sphérique. Avec des paramètres globaux, l'ajustement des cylindres sur ce sous-maillage aurait été rejeté à cause des imprécisions trop importantes.

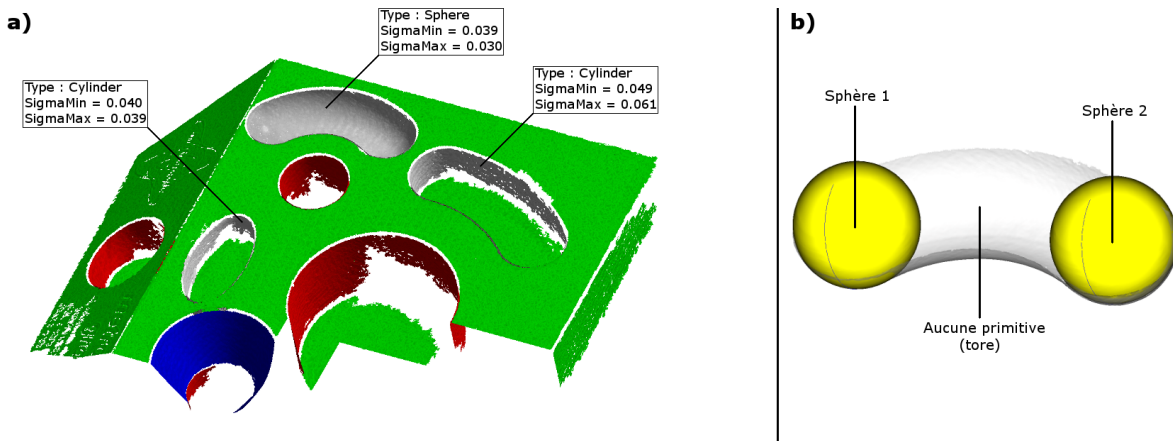


FIGURE 3.33 – Analyse des sous-maillages de type inconnu de *Metrologic*: a) Type majoritaire et tolérances, b) Primitives ajustées localement sur un des sous-maillages.

3.6 Résultats expérimentaux

Dans cette section, nous comparons l'extraction de primitives sur plusieurs maillages, avec la méthode de Bénéière *et al.* [5] et avec la méthode proposée composée de deux segmentations, deux ajustements globaux et un ajustement local (voir section 3.1). A partir de *Block*, la méthode de Bénéière *et al.* [5] permet d'extraire 5 plans couvrant 42,92% du maillage (Fig. 3.34.a). Avec notre méthode, nous obtenons 91 primitives représentant 99,35% du maillage (Fig. 3.34.b). Ces primitives permettent alors de reconstruire l'objet dans sa totalité, puisque les seules parties du maillage non couvertes par des primitives correspondent aux arêtes vives.

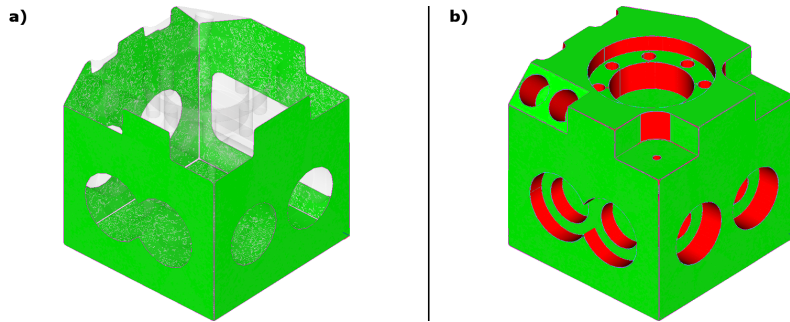


FIGURE 3.34 – *Block* : primitives extraites avec a) la méthode de Bénéière *et al.* [5] et b) la nôtre.

La figure 3.35 montre les primitives extraites à partir du maillage *Lego*. Avec la méthode de Bénéière *et al.* [5], seuls deux plans sont extraits (Fig. 3.35.a). L'aire des sous-maillages associés ne représente alors que 7,13% de l'aire totale du maillage. Avec notre méthode, 25 primitives sont extraites et représentent 85,12% du maillage (Fig. 3.35.b). Il est à noter que nous ne prenons pas en compte les tores dans notre processus, d'où les zones manquantes au niveau des quatre plots.

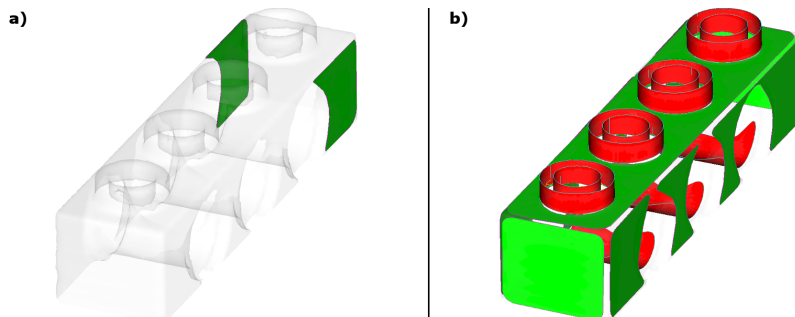


FIGURE 3.35 – *Lego* : primitives extraites avec a) la méthode de Bénéière *et al.* [5] et b) la nôtre.

Lorsqu'un maillage est trop imprécis ou complexe, il est fréquent de n'extraire aucune primitive avec la méthode de Bénéière *et al.* [5], comme nous pouvons le voir sur la figure 3.36.a. Grâce à nos adaptations, nous pouvons extraire l'ensemble des primitives de *Gear* (mis à part les parties toriques). Ainsi, 84,04% du maillage est couvert par des primitives (Fig. 3.36.b).

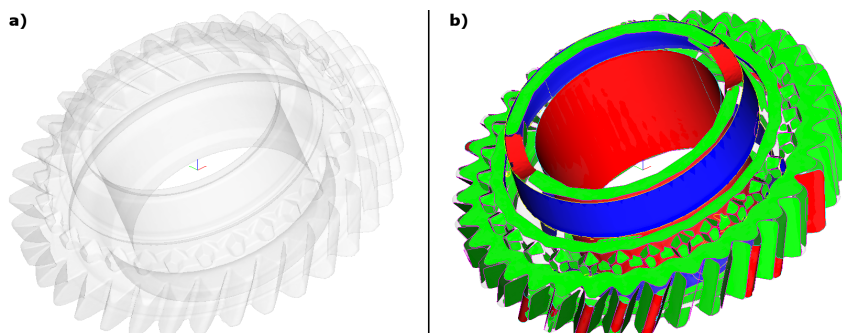


FIGURE 3.36 – *Gear* : primitives extraites avec a) la méthode de Bénéière *et al.* [5] et b) la nôtre.

Nous avons réalisé une analyse similaire sur notre sélection de 30 objets. Le tableau 3.6 présente les performances de l'extraction de primitives à chaque étape, en termes de temps de calcul et de couverture du maillage par des primitives. Cette couverture est calculée par un ratio entre l'aire cumulée des sous-maillages sous-jacents aux primitives, sans chevauchement, et l'aire totale du maillage. Ces résultats montrent que chaque étape permet d'augmenter le taux de couverture tout en gardant des temps de calcul raisonnables.

Avec la méthode de Bénéière *et al.* [5], nous obtenons une couverture moyenne de 12,5% et une couverture médiane de 0%. En effet, de nombreux maillages sont trop imprécis pour extraire des primitives à l'aide de cette méthode, originalement conçue pour les maillages issus de CAO. Grâce à la méthode proposée, nous obtenons au final une couverture moyenne de 77,3% et une couverture médiane de 79,4%. Il est à noter que ces pourcentages ne peuvent pas atteindre 100% puisque nous n'intégrons pas les régions arêtes. Ces statistiques ont été calculées à partir de tous les objets, y compris ceux qui sont en dehors du cadre classique de rétro-ingénierie, comme par exemple un vase ou une chaussure. Enfin, plusieurs objets contiennent des tores, que nous ne cherchons pas à extraire ici, ou encore des surfaces à forme libre, non représentables par des primitives.

En ne considérant que les maillages correctement numérisés et qui contiennent uniquement des primitives simples, notés dans le tableau 3.6 par un astérisque, nous obtenons une couverture moyenne de 92,4% (contre 31,7% avec la méthode de Bénéière *et al.* [5]). Dans l'ensemble, ces résultats sont donc très satisfaisants et mettent en avant la robustesse de la méthode proposée.

Maillage	Temps (en secondes)				Couverture du maillage (en %)			
	Bénéière <i>et al.</i> [5]	Notre méthode			Bénéière <i>et al.</i> [5]	Notre méthode		
		(a)	(b)	(c)		(a)	(b)	(c)
* Fandisk	1	1	1	1	86.66	45.83	76.02	90.77
Vase	1	1	1	1	4.23	9.97	31.40	31.83
* Lego	1	1	1	1	7.13	78.04	84.66	85.43
* Lego_small	1	1	1	1	70.94	96.27	96.27	96.27
Cup	1	2	3	3	0.00	2.98	68.99	91.26
* Manique	1	3	3	4	45.14	49.80	88.18	92.42
* Nespresso	1	2	2	2	46.55	96.73	96.73	96.73
MediumBolt	1	4	5	5	0.00	12.01	73.91	78.45
* Hydrant	1	3	4	4	0.00	36.91	78.68	85.54
StripedShoe	1	3	6	7	0.00	3.52	25.92	64.38
Prise	2	4	5	5	0.00	70.24	71.67	76.56
Connecteur	2	5	7	9	0.00	12.83	49.25	82.10
Spicelid	2	6	7	9	0.00	44.29	67.95	71.64
* Etui	4	8	9	9	17.26	81.84	81.84	93.23
Blade	7	9	12	18	0.00	51.42	51.42	77.96
Czslowakei	7	15	22	32	0.00	16.10	26.89	37.26
Chair	7	15	21	23	0.00	52.14	63.63	77.92
Gear	8	16	29	31	0.00	49.41	83.26	84.04
Aerospace	23	18	28	31	0.00	39.77	68.93	72.42
Master	22	48	94	114	0.00	1.97	54.85	73.99
Moldy	20	19	30	64	51.58	79.42	80.20	81.28
Watertight	32	34	54	88	0.00	20.04	40.07	62.27
Aube	38	31	44	52	1.26	28.56	47.06	49.78
OilPump	34	58	91	99	0.00	28.16	73.58	79.30
Carter	40	51	56	61	0.00	57.93	60.06	62.23
Pump	44	64	87	98	0.00	56.50	71.99	76.72
* Block	41	35	37	38	42.92	92.37	97.37	99.35
Transmission	51	95	264	335	0.00	17.56	64.95	79.45
* Splint	113	59	88	119	0.00	10.47	73.42	98.46
* Metrologic	120	139	147	164	0.00	80.73	81.19	85.81

TABLEAU 3.6 – Comparatif de l'extraction de primitives avec la méthode de Bénéière *et al.* [5] et la nôtre : a) Après la segmentation "arêtes vives" et l'ajustement global, b) Après la segmentation "arêtes lisses" et le second ajustement global, c) Après l'ajustement local avec calcul automatique des paramètres.

Nous avons ensuite quantifié manuellement les primitives correctes et incorrectes obtenues. Nous pouvons voir dans le tableau 3.7 que chaque étape permet d'augmenter le nombre de primi-

tives correctes. D'un autre côté, certaines primitives peuvent être incorrectes, c'est-à-dire jugées comme ne faisant pas partie du modèle CAO original par une estimation visuelle. En effet, les primitives sont extraites indépendamment dans les différentes étapes et aucune vérification ne permet de mettre en avant une quelconque logique de construction d'un objet. En fait, la plupart des primitives jugées incorrectes se situent dans les zones très bruitées des objets, et notamment aux abords de leurs arêtes vives.

Premièrement, la segmentation "arêtes vives" permet généralement d'extraire un grand nombre de primitives par ajustement global. En effet, comme décrit dans la section 3.2.4, la majorité des sous-maillages obtenus ne contiennent qu'une seule primitive. Cependant, certains sous-maillages n'ont pas de cohérence du point de vue conception de la pièce. Dans ce cas, les primitives correspondantes peuvent être jugées incorrectes. Deuxièmement, la segmentation "arêtes lisses" permet de dissocier de nombreuses primitives tangentes, comme détaillé dans la section 3.4, qui sont par la suite extraites par ajustement global. Troisièmement, l'ajustement local permet d'extraire des primitives supplémentaires en complétant au maximum la couverture du maillage. Cependant, celui-ci est réalisé sur tous les sous-maillages de type inconnus. Des primitives sont alors extraites à partir de sous-maillages correspondant, par exemple, à des surfaces complexes. Dans ce cas, des primitives sont jugées incorrectes. Mais ces primitives incorrectes correspondent généralement à des sous-maillages de peu de points, et donc une faible couverture du maillage, comme c'est le cas pour le maillage *Metrologic* par exemple. Au final, le processus d'extraction en cinq étapes proposé est robuste et permet d'extraire de nombreuses primitives.

Maillage	Nombre de primitives correctes / incorrectes			Couverture du maillage (en %) correcte / incorrecte		
	(a)	(b)	(c)	(a)	(b)	(c)
Fandisk	8 / 0	17 / 0	22 / 0	45.83 / 0.00	76.02 / 0.00	90.77 / 0.00
Vase	3 / 0	6 / 6	8 / 5	9.97 / 0.00	26.44 / 4.96	27.17 / 4.21
Lego	23 / 1	24 / 1	25 / 1	77.73 / 0.31	84.35 / 0.31	85.12 / 0.31
Lego_small	9 / 0	9 / 0	9 / 0	96.27 / 0.00	96.27 / 0.00	96.27 / 0.00
Cup	8 / 0	18 / 26	32 / 30	2.98 / 0.00	50.51 / 18.48	70.29 / 20.97
Manique	12 / 0	44 / 34	54 / 46	49.80 / 0.00	80.98 / 7.20	81.64 / 10.78
Nespresso	4 / 1	4 / 1	4 / 1	96.55 / 0.18	96.55 / 0.18	96.55 / 0.18
MediumBolt	3 / 0	26 / 148	33 / 168	12.01 / 0.00	50.48 / 23.43	52.64 / 25.81
Hydrant	61 / 6	84 / 32	88 / 58	35.80 / 1.11	71.75 / 6.93	73.64 / 11.90
StripedShoe	4 / 8	8 / 53	14 / 75	1.74 / 1.78	4.46 / 21.46	18.17 / 46.21
Prise	60 / 15	60 / 22	66 / 22	66.89 / 3.35	66.99 / 4.68	71.32 / 5.24
Connector	21 / 3	41 / 18	51 / 21	12.59 / 0.24	47.00 / 2.25	74.17 / 7.93
Spicelid	47 / 17	68 / 58	72 / 64	42.88 / 1.41	63.00 / 4.95	65.43 / 6.21
Etui	2 / 4	2 / 4	3 / 7	81.25 / 0.59	81.25 / 0.59	87.46 / 5.77
Blade	40 / 38	40 / 38	48 / 42	46.56 / 4.86	46.56 / 4.86	68.79 / 9.17
Czslowakei	107 / 3	136 / 13	139 / 24	15.91 / 0.19	24.67 / 2.12	28.98 / 8.28
Chair	60 / 46	79 / 88	89 / 105	49.99 / 2.15	58.98 / 4.65	72.76 / 5.16
Gear	63 / 13	305 / 52	308 / 62	48.67 / 0.74	80.19 / 3.07	80.44 / 3.60
Aerospace	26 / 0	60 / 31	69 / 36	39.77 / 0.00	66.95 / 1.98	69.56 / 2.86
Master	2 / 61	81 / 507	117 / 652	0.14 / 1.83	36.09 / 18.76	42.49 / 31.50
Moldy	26 / 0	29 / 4	35 / 4	79.42 / 0.00	79.66 / 0.54	80.41 / 0.87
Watertight	49 / 74	95 / 168	132 / 209	17.35 / 2.69	33.51 / 6.56	53.10 / 9.17
Aube	32 / 8	53 / 10	53 / 24	28.21 / 0.35	46.56 / 0.50	46.56 / 3.22
OilPump	104 / 242	190 / 443	230 / 472	21.56 / 6.60	61.28 / 12.30	65.30 / 14.00
Carter	48 / 244	48 / 252	63 / 262	49.96 / 7.97	49.96 / 10.10	53.62 / 8.61
Pump	449 / 66	599 / 141	745 / 187	55.29 / 1.21	69.42 / 2.57	72.66 / 4.06
Block	64 / 0	85 / 0	91 / 0	92.37 / 0.00	97.37 / 0.00	99.35 / 0.00
Transmission	401 / 27	1904 / 117	2306 / 232	17.26 / 0.30	62.97 / 1.98	75.71 / 3.74
Splint	7 / 6	10 / 6	23 / 9	10.42 / 0.05	73.37 / 0.05	79.06 / 1.40
Metrologic	60 / 307	60 / 310	64 / 314	72.60 / 8.13	72.60 / 8.59	74.87 / 10.94

TABEAU 3.7 – Analyse des primitives obtenues avec notre méthode : a) Après la segmentation "arêtes vives" et l'ajustement global, b) Après la segmentation "arêtes lisses" et le second ajustement global, c) Après l'ajustement local avec calcul automatique des paramètres.

3.7 Conclusion

Dans ce chapitre, un processus en cinq étapes a été proposé dans le but d'améliorer la robustesse de l'extraction des primitives géométriques à partir d'un maillage 3D issu d'une numérisation.

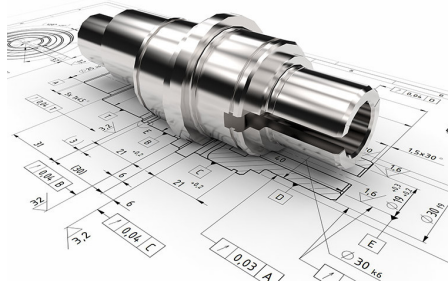
Dans un premier temps, une segmentation est réalisée à l'aide d'une analyse des distributions de la courbure moyenne permettant de discriminer les points du maillage appartenant aux arêtes vives de l'objet. Celle-ci permet, dans 96% des cas en moyenne, de séparer correctement les régions du maillage correspondant aux différentes primitives. Ensuite, un ajustement global des primitives est réalisé sur chaque sous-maillage. Ce dernier permet principalement un gain en précision et en temps de calcul, en prenant en compte des sous-ensembles de points importants. Cependant, certains sous-maillages peuvent encore contenir plusieurs primitives après la segmentation, notamment dans le cas de primitives voisines par des arêtes lisses (4% des cas). Dans ce cas, aucune primitive unique ne peut être ajustée correctement et le sous-maillage est de type inconnu. Une seconde segmentation est alors réalisée sur les sous-maillages de type inconnu, afin de séparer les régions correspondant à des primitives séparées par des arêtes lisses. Celle-ci repose également sur une analyse des distributions des courbures, et permet de réduire le taux de sous-maillages de type inconnu à 0,3% en moyenne. Ensuite, un second ajustement global est réalisé sur les sous-maillages obtenus. Enfin, un ajustement local est réalisé sur les sous-maillages restants, de type inconnu, par croissance de régions sur les valeurs des courbures des points. Cette dernière étape repose sur une analyse des distributions des courbures permettant de calculer automatiquement les tolérances utilisées lors de la croissance de régions. Ainsi, le processus est capable de s'adapter en fonction de la répartition des imprécisions indépendamment pour chaque sous-maillage.

Au final, nous avons vu à travers des résultats expérimentaux que notre méthode permet l'extraction de nombreuses primitives, même en présence d'imprécisions importantes. De plus, ces primitives sont en général suffisamment fiables, puisqu'elles sont estimées à partir d'un grand nombre de points. Enfin, les différentes étapes proposées permettent de réduire significativement les temps de calcul. En effet, les étapes de segmentation permettent de traiter des sous-maillages plus restreints, et les étapes d'ajustement global permettent d'éviter le calcul des courbures qui est souvent coûteux. Il est alors possible de traiter des maillages de plusieurs millions de points en quelques minutes.

Les travaux présentés dans ce chapitre ont été valorisés par les publications suivantes :

- Gauthier, S., Puech, W., Bénière, R., Subsol, G. (2016). Segmentation de maillages 3D de pièces manufacturées numérisées : Application à la rétro-conception. *CORESA*, 2016(18) : 40-45.
- Gauthier, S., Puech, W., Bénière, R., Subsol, G. (2017). Digitized 3D mesh segmentation based on curvature analysis. *Electronic Imaging*, 2017(20), 33-38.
- Gauthier, S., Puech, W., Bénière, R., Subsol, G. (2017). Analysis of digitized 3D mesh curvature histograms for reverse engineering. *Computers in Industry*, 92, 67-83.

Cependant, certaines primitives extraites peuvent être jugées incorrectes si l'on se réfère aux modèles CAO originaux. En effet, les primitives sont extraites indépendamment et aucune vérification n'est réalisée pour suivre une logique de conception. Il serait alors intéressant de chercher à retrouver des informations plus globales, qui pourraient alors permettre de corriger certaines primitives. Le chapitre 4 tente d'apporter quelques éléments de réponse à cette nouvelle problématique.



Chapitre 4

Analyse sémantique et embellissement du modèle reconstruit

« Cherchez la raison de toute forme, car toute forme a sa raison. »

Eugène Viollet-le-Duc

Résumé

Dans le chapitre 3, nous avons vu que les primitives extraites d'un maillage numérisé nécessitent parfois d'être corrigées. Comme décrit dans le chapitre 2, l'analyse d'un ensemble de primitives consiste souvent à extraire deux types d'information : un système de référence CAO et des motifs particuliers. Il semble cependant qu'aucune méthode actuelle ne soit réellement adaptée à notre problématique, c'est-à-dire de pouvoir traiter efficacement des primitives dont les paramètres ne sont pas "exact". Nous proposons alors une méthode d'embellissement basée sur une analyse globale et robuste permettant d'extraire des informations pour régulariser les primitives.

Après une brève introduction (section 4.1), nous présentons dans la section 4.2 la construction d'un graphe de relations géométriques entre les primitives. Dans la section 4.3, nous décrivons un embellissement des primitives basé sur une analyse globale et une régularisation en se basant sur l'estimation d'un système de référence CAO. Nous détaillons ensuite un algorithme de reconnaissance de motifs particuliers dans la section 4.4. Enfin, nous présentons des résultats expérimentaux (section 4.5), suivis d'une conclusion (section 4.6).

Sommaire

4.1 Introduction	92
4.2 Construction d'un graphe de relations entre les primitives	93
4.2.1 Relations d'adjacence	93
4.2.2 Orientations, positions et dimensions relatives	94
4.2.3 Dédution de relations complexes	96
4.2.4 Convexité et concavité	96
4.3 Analyse globale et embellissement des primitives	98
4.3.1 Analyse des orientations des primitives	98
4.3.2 Analyse des positions des primitives	101
4.3.3 Analyse des dimensions des primitives	104
4.3.4 Résultats expérimentaux	104
4.4 Reconnaissance de motifs particuliers	111
4.4.1 Entités géométriques	111
4.4.2 Répétitions de géométrie	113
4.4.3 Application à l'embellissement	119
4.4.4 Résultats expérimentaux	121
4.5 Expérience d'embellissement intégré à la chaîne de traitement	123
4.6 Conclusion	128

4.1 Introduction

Afin de répondre aux problématiques mises en avant dans les chapitres 2 et 3, nous proposons une méthode d'embellissement, qui cherche à régulariser les paramètres des primitives (voir section 2.4). Celle-ci se décompose en quatre étapes (Fig. 4.1) : une analyse globale des primitives, un embellissement, une reconnaissance de motifs et un second embellissement.

Ces quatre étapes ont pour objectif d'améliorer le modèle reconstruit, afin que celui-ci soit valide et corresponde au mieux au modèle initial. Il semble important de noter que les informations sémantiques extraites reflètent un processus de modélisation CAO de la pièce originale.

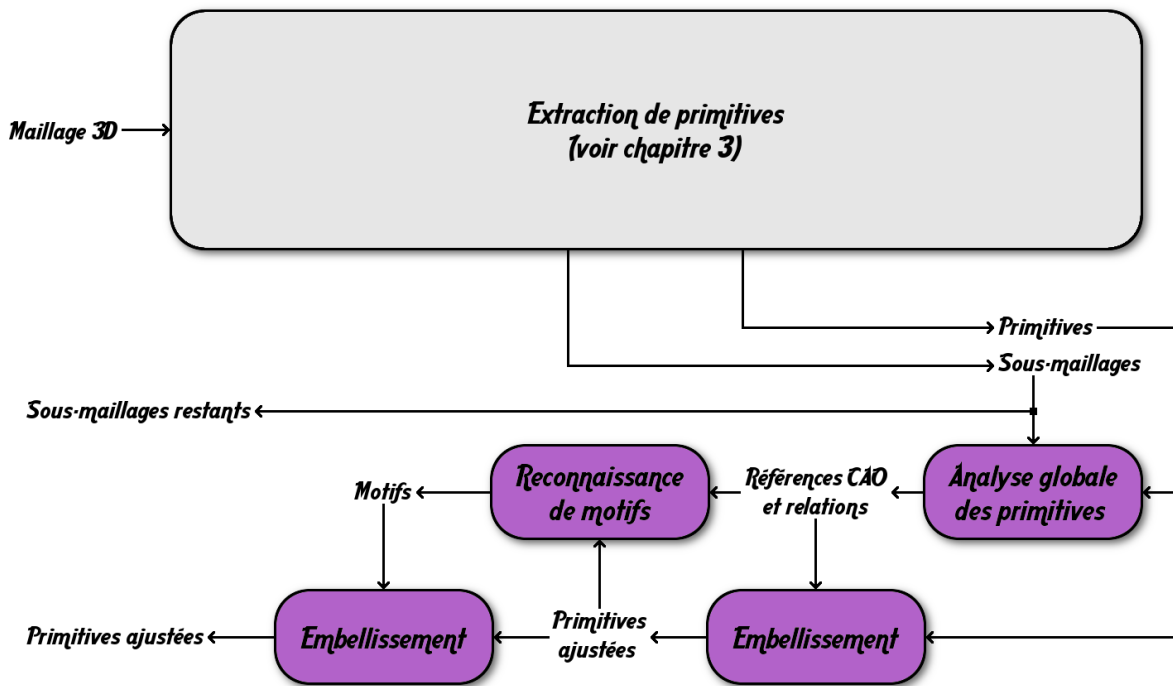


FIGURE 4.1 – Schéma global de notre approche avec analyse et embellissement des primitives.

L'analyse globale des primitives est basée sur la construction d'un graphe de relations géométriques telles que le parallélisme, l'orthogonalité et la concentricité. Ces relations sont d'abord déterminées entre chaque paire de primitives en fonction de leurs paramètres respectifs. Ensuite, un graphe est construit en définissant un noeud pour chaque primitive et un arc pour chaque relation obtenue. Enfin, ce graphe est analysé afin d'en déduire un système de référence CAO correspondant à l'objet. Cette étape est décrite dans la section 4.2.

L'embellissement consiste à régulariser les paramètres des primitives en fonction du système de référence CAO déduit du graphe de relations, afin qu'elles respectent exactement des relations géométriques. En effet, le système de référence CAO permet de régulariser chaque primitive indépendamment, en évitant une éventuelle propagation d'erreur. De plus, ce processus respecte des règles inhérentes au processus de modélisation CAO, garantissant une meilleure fidélité avec le modèle initial. Cette étape est détaillée dans la section 4.3.

La reconnaissance de motifs est une seconde phase d'analyse, par sous-ensembles de primitives. Cette dernière a pour objectif de retrouver des configurations particulières de primitives, appelés motifs, en fonction des standards de modélisation CAO. Ainsi, certains motifs correspondent par exemple à des *entités* ou des *répétitions* (voir section 2.3). Cette étape est présentée dans la section 4.4. En particulier, les motifs extraits sont par la suite utilisés pour une seconde étape d'embellissement, comme décrit dans la section 4.4.3.

4.2 Construction d'un graphe de relations entre les primitives

La modélisation d'une pièce manufacturée en CAO implique presque toujours des règles de construction comme le parallélisme, l'orthogonalité ou la concentricité. Or, ces relations ne sont pas explicitement présentes dans l'ensemble des primitives extraites à partir d'un maillage 3D. De plus, les paramètres des primitives peuvent être imprécis. Pour identifier ces relations géométriques, il est donc nécessaire de définir un système de tolérances. Nous définissons alors un ensemble de tolérances géométriques :

- $\Delta_{distance}$: tolérance de distance permettant de définir deux positions similaires ;
- Δ_{angle} : tolérance d'angle permettant de définir deux orientations similaires ;
- Ω_{angles} : ensemble d'angles particuliers définis entre deux orientations.

4.2.1 Relations d'adjacence

Dans un premier temps, nous cherchons à construire la topologie du modèle, c'est à dire un ensemble des relations d'adjacence entre les primitives. Pour rappel, la méthode d'extraction de primitives du chapitre 3 nous fournit trois types de données :

- des primitives, chacune composée d'un sous-maillage support et d'une surface définie par un ensemble de paramètres ;
- des sous-maillages restants, de type inconnu, sur lesquels aucune primitive n'a été ajustée ;
- des sous-maillages arêtes, qui ont été retirés lors des deux étapes de segmentation.

Nous proposons alors de propager (ou dilater) les sous-maillages support des primitives, sur les sous-maillages arêtes et de manière synchronisée, jusqu'à ce qu'ils entrent en collision (Fig. 4.2). Ainsi, une relation d'adjacence est détectée lorsque deux sous-maillages supports entrent en collision. Cela permet alors de construire un graphe d'adjacence, qui peut être utilisé par la suite pour renforcer d'autres relations entre les primitives voisines ou encore détecter des entités (voir section 4.4).

Afin d'éviter les incohérences dans ce graphe, la propagation est interdite sur les sous-maillages restants. Des relations spécifiques d'adjacence sont alors définies entre les sous-maillages supports et les sous-maillages restants qui entrent en collision.

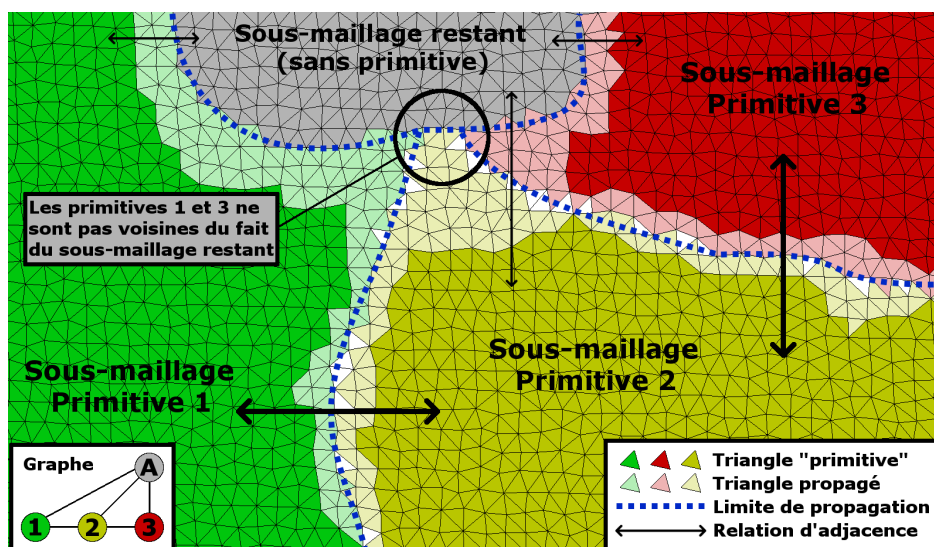


FIGURE 4.2 – Adjacence des primitives par propagation de leur sous-maillage support, sur les sous-maillages arêtes et de manière synchronisée. L'adjacence est déterminée par les collisions.

4.2.2 Orientations, positions et dimensions relatives

Les relations géométriques entre les primitives sont définies par leurs orientations, positions et dimensions respectives. Pour rappel, une primitive géométrique est définie par un ensemble de paramètres :

- un plan est défini par une normale et un point, d'où $\text{Plan}_i = \{\vec{n}_i, P_i\}$;
- une sphère est définie par un rayon et un point (centre), d'où $\text{Sphère}_i = \{r_i, P_i\}$;
- un cylindre est défini par un rayon, un vecteur directeur et un point, d'où $\text{Cylindre}_i = \{r_i, \vec{n}_i, P_i\}$;
- un cône est défini par un demi-angle, un vecteur directeur et un point (sommet), d'où $\text{Cône}_i = \{\psi_i, \vec{n}_i, P_i\}$.

Nous proposons d'analyser ces paramètres le plus globalement possible, afin d'atténuer l'impact des imprécisions des primitives extraites à partir d'un maillage issu de numérisation. Pour cela, il est nécessaire de mettre en place un système de regroupement des paramètres similaires.

Regroupement des orientations

Premièrement, nous cherchons à construire des groupes d'orientations similaires à Δ_{angle} près. Ainsi, les normales aux plans et les vecteurs directeurs des axes des cylindres et des cônes forment un ensemble d'orientations $\{\vec{V}_0 \dots \vec{V}_n\}$. Pour les regrouper, nous réalisons trois étapes, "naïves" mais suffisantes dans notre cas, jusqu'à ce que toutes les orientations soient traitées.

1. Créer un nouveau groupe avec une orientation \vec{V}_i qui n'est pas encore regroupée.
2. Ajouter au groupe toutes les orientations \vec{V}_j telles que $\widehat{\vec{V}_i, \vec{V}_j} \leq \Delta_{angle}$.
3. Répéter l'étape 2 pour chaque \vec{V}_j , jusqu'à ne plus pouvoir ajouter de nouvelle orientation.

Nous calculons ensuite une orientation moyenne $\vec{\mathcal{O}}_k$ et un poids total ω_k pour chaque groupe :

$$\vec{\mathcal{O}}_k = \frac{1}{\omega_k} \sum_{i=1}^n \omega_i \vec{V}_i \quad \text{avec} \quad \omega_k = \sum_{i=1}^n \omega_i, \quad (4.1)$$

où \vec{V}_i est une des n orientations du groupe et ω_i le poids associé à \vec{V}_i , correspondant à l'aire du sous-maillage de la primitive associée. Cet algorithme regroupe donc les orientations parallèles entre elles, en fonction de Δ_{angle} . Dans la suite, les orientations moyennes $\vec{\mathcal{O}}_k$ sont normalisées.

Nous pouvons alors définir une relation de parallélisme entre chaque paire de primitives appartenant au même groupe. De même, nous pouvons calculer les angles formés par chaque paire d'orientations moyennes $(\vec{\mathcal{O}}_i, \vec{\mathcal{O}}_j)$ afin de définir des relations d'orientations entre des primitives de groupes différents. En particulier, une relation d'orientation est définie lorsque :

$$\alpha - \Delta_{angle} \leq \widehat{\vec{\mathcal{O}}_i, \vec{\mathcal{O}}_j} \leq \alpha + \Delta_{angle} \quad \text{avec} \quad \alpha \in \Omega_{angles}. \quad (4.2)$$

Notons en particulier que l'orthogonalité est définie par $\alpha = 90^\circ$.

Regroupement des positions

Deuxièmement, nous cherchons à construire des groupes de positions similaires à $\Delta_{distance}$ près. Ainsi, les centres des sphères forment un ensemble de positions $\{P_0 \dots P_k\}$. Un second processus de regroupement est alors réalisé, ce qui permet de définir des relations de concentricité entre les sphères appartenant au même groupe :

1. Créer un groupe avec une position P_i qui n'est pas encore regroupée, prise au hasard.
2. Ajouter au groupe toutes les positions P_j telles que $\|\vec{P}_j - \vec{P}_i\| \leq \Delta_{distance}$.
3. Répéter l'étape 2 pour chaque P_j , jusqu'à ne plus pouvoir ajouter de nouvelle position.

Dans le cas des cylindres et des cônes, les relations de concentricité impliquent des axes similaires. Or, ces derniers sont définis par une position et une orientation. Il est alors nécessaire de ne considérer que des cylindres et cônes parallèles, c'est à dire appartenant au même groupe d'orientations. Ensuite, les positions pouvant être définies n'importe où sur les axes, une distance point à point n'est pas cohérente. Pour mesurer la distance entre deux axes et la comparer à $\Delta_{distance}$, nous prenons dans ce cas la position d'un axe et sa projection sur l'autre axe.

Dans le cas des plans, nous pouvons définir des relations de coplanarité. Comme pour les cylindres et les cônes, nous ne considérons que les plans parallèles. Ensuite, nous calculons la distance entre le point d'un plan et son projeté sur un second plan pour décider s'ils sont similaires.

Comme pour les orientations, nous pouvons définir une position moyenne \mathcal{P}_k et un poids total ω_k pour chaque groupe :

$$\mathcal{P}_k = \frac{1}{\omega_k} \sum_{i=1}^n \omega_i P_i \quad \text{avec} \quad \omega_k = \sum_{i=1}^n \omega_i, \quad (4.3)$$

où P_i est une des n positions du groupe et ω_i le poids associé à P_i , correspondant à l'aire de la primitive géométrique associée.

Dans le cas d'un regroupement de plusieurs cylindres et cônes, l'ensemble des positions $\{P_1 \dots P_n\}$ correspond aux projections d'un point arbitraire sur les n axes correspondant. Dans le cas d'un regroupement de plusieurs plans, ce point arbitraire est projeté directement sur les primitives.

Nous pouvons alors définir des relations de coplanarité et de concentricité entre les primitives en s'appuyant sur les fonctions de distance présentées dans la section 3.3. Voici une liste des conditions de coplanarité et de concentricité entre deux primitives.

- Coplanarité Plan₁-Plan₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{plan}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$.
- Concentricité Sphère₁-Sphère₂ : $\|\vec{P_1 P_2}\| \leq \Delta_{distance}$.
- Concentricité Sphère₁-Cylindre₂ : $\mathcal{D}_{axe}(\vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$.
- Concentricité Sphère₁-Cône₂ : $\mathcal{D}_{axe}(\vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$.
- Concentricité Cylindre₁-Cylindre₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{axe}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$.
- Concentricité Cylindre₁-Cône₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{axe}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$.
- Concentricité Cône₁-Cône₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{axe}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$.

Bien que nous ayons utilisé les tolérances Δ_{angle} et $\Delta_{distance}$, il est possible de définir des tolérances différentes pour chaque paire de primitive.

Regroupement des dimensions

Troisièmement, nous cherchons à regrouper les primitives ayant des dimensions similaires. Ainsi, les rayons des sphères et des cylindres sont regroupés en fonction de $\Delta_{distance}$, et les demi-angles des cônes en fonction de Δ_{angle} . Comme précédemment, des rayons moyens \mathcal{R}_k et des demi-angles moyens \mathcal{A}_k sont calculés par :

$$\mathcal{R}_k = \frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i r_i \quad \text{ou} \quad \mathcal{A}_k = \frac{1}{\sum_{i=1}^n \omega_i} \sum_{i=1}^n \omega_i \psi_i, \quad (4.4)$$

où r_i est un rayon d'un cylindre ou d'une sphère et ψ_i est un demi-angle d'un cône.

Nous pouvons alors définir des relations d'égalité dimensionnelle entre les primitives issues du même groupe. Plus généralement, il est possible de définir des relations de proportionnalité entre deux rayons (r_i, r_j) ou deux demi-angles (ψ_i, ψ_j) si :

$$\frac{\max(r_i; r_j)}{\min(r_i; r_j)} = K_r \quad \text{ou} \quad \frac{\max(\psi_i; \psi_j)}{\min(\psi_i; \psi_j)} = K_\psi, \quad (4.5)$$

où K_r et K_ψ sont des coefficients donnés.

4.2.3 Dédution de relations complexes

Nous pouvons aussi chercher à détecter des relations plus complexes entre les primitives, correspondant à des combinaisons de relations d'orientations, de positions et de dimensions. Même s'il est possible de les détecter entre des primitives éloignées, dans notre étude, ces relations sont généralement définies entre des primitives voisines.

Ainsi, nous cherchons à déterminer des relations de tangence. Cette détection étant spécifique aux types des primitives mises en jeu, l'algorithme doit être relativement exhaustif. Cependant, certaines tangences sont non-triviales, notamment celles mettant en jeu des cônes. Dans ces travaux, nous nous concentrons essentiellement sur les cas de tangence les moins complexes, qui sont souvent les plus fréquents. En particulier, ces tangences concernent souvent des congés.

Voici une liste des conditions de tangence prises en compte ici :

- Plan₁-Sphère₂ : $|\mathcal{D}_{plan}(\vec{n}_1, P_1, P_2)| = r_2 \pm \Delta_{distance}$
- Plan₁-Cylindre₂ : $\widehat{n_1 n_2} = 90 \pm \Delta_{angle}$ **et** $|\mathcal{D}_{plan}(\vec{n}_1, P_1, P_2)| = r_2 \pm \Delta_{distance}$
- Plan₁-Cône₂ : $\widehat{n_1 n_2} = \psi_2 \pm \Delta_{angle}$ **et** $\mathcal{D}_{plan}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$
- Sphère₁-Sphère₂ : $|\mathcal{D}_{sphère}(r_1, P_1, P_2)| = r_2 \pm \Delta_{distance}$
- Sphère₁-Cylindre₂ : $|\mathcal{D}_{cylindre}(r_2, \vec{n}_2, P_2, P_1)| = r_1 \pm \Delta_{distance}$
- Sphère₁-Cône₂ : $|\mathcal{D}_{cône}(0, \psi_2, \vec{n}_2, P_2, P_1)| = r_1 \pm \Delta_{distance}$
- Cylindre₁-Cylindre₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $|\mathcal{D}_{cylindre}(r_1, \vec{n}_1, P_1, P_2)| = r_2 \pm \Delta_{distance}$
- Cylindre₁-Cône₂ : $\widehat{n_1 n_2} = \psi_2 \pm \Delta_{angle}$ **et** $|\mathcal{D}_{cylindre}(r_1, \vec{n}_1, P_1, P_2)| = 0 \pm \Delta_{distance}$

Enfin, nous définissons également une relation dérivée de la tangence, que nous nommons *coïncidence*. Celle-ci est principalement définie lorsqu'une surface contient l'axe ou le centre d'une autre primitive. Ainsi, nous proposons quelques conditions de coïncidence :

- Sphère₁-Plan₂ : $\mathcal{D}_{plan}(\vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Sphère₁-Sphère₂ : $\mathcal{D}_{sphère}(r_1, P_1, P_2) = 0 \pm \Delta_{distance}$
- Sphère₁-Cylindre₂ : $\mathcal{D}_{cylindre}(r_2, \vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Sphère₁-Cône₂ : $\mathcal{D}_{cône}(\psi_2, \vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Cylindre₁-Plan₂ : $\widehat{n_1 n_2} = 90 \pm \Delta_{angle}$ **et** $\mathcal{D}_{plan}(\vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Cylindre₁-Cylindre₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{cylindre}(r_1, \vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$
- Cylindre₁-Cône₂ : $\widehat{n_1 n_2} = \psi_2 \pm \Delta_{angle}$ **et** $\mathcal{D}_{axe}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$
- Cône₁-Plan₂ : $\widehat{n_1 n_2} = 90 \pm \Delta_{angle}$ **et** $\mathcal{D}_{plan}(\vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Cône₁-Cylindre₂ : $\widehat{n_1 n_2} = 0 \pm \Delta_{angle}$ **et** $\mathcal{D}_{cylindre}(r_2, \vec{n}_2, P_2, P_1) = 0 \pm \Delta_{distance}$
- Cône₁-Cône₂ : $\widehat{n_1 n_2} = \psi_2 \pm \Delta_{angle}$ **et** $\mathcal{D}_{axe}(\vec{n}_1, P_1, P_2) = 0 \pm \Delta_{distance}$

Notons que contrairement aux relations précédentes, la coïncidence est orientée. Par exemple, un cylindre peut contenir le centre d'une sphère alors qu'une sphère ne peut contenir l'axe d'un cylindre. De même, le plan ne peut pas coïncider avec une autre primitive.

4.2.4 Convexité et concavité

Les notions de convexité et de concavité permettent une analyse plus fine des ensembles de primitives. Nous définissons ici une primitive comme convexe (resp. concave) lorsqu'elle délimite l'extérieur (resp. l'intérieur) de l'objet. Ainsi, un cylindre concave suggère un perçage ou un évidement, tandis qu'un cylindre convexe suggère plutôt un arbre ou un épaulement. Dans un premier temps, nous estimons la convexité d'une primitive à l'aide du sous-maillage correspondant. Ensuite, nous définissons une relation entre deux primitives représentant la convexité de leur intersection. Cette dernière permet par exemple de détecter des positions relatives telles que des inclusions ou des coïncidences de primitives, c'est-à-dire l'une dans l'autre ou l'une sur l'autre.

Convexité des primitives

Pour définir la convexité d'une primitive, nous nous appuyons sur les paramètres définissant sa surface et les normales du sous-maillage sous-jacent. Dans le cas d'un plan, il n'y a pas de notion de convexité. Pour les autres primitives, l'opération consiste à comparer les normales du sous-maillage aux normales théoriques sur la primitive. Si ces normales sont globalement dans le même sens, nous pouvons en déduire que la primitive est convexe, et inversement pour une primitive concave.

Ainsi, nous proposons de calculer un indice de convexité I_c à partir d'un point de référence P , correspondant au centre d'une sphère ou au point de l'axe d'un cylindre ou d'un cône :

$$I_c = \frac{1}{\omega} \sum_{i=1}^n \omega_i \frac{s_i}{|s_i|} \quad \text{avec} \quad \omega = \sum_{i=1}^n \omega_i \quad \text{et} \quad s_i = \vec{n}_i \cdot \overrightarrow{PP_i}, \quad (4.6)$$

où n est le nombre de points P_i du sous-maillage, ω_i le poids associé à P_i et \vec{n}_i la normale sortante en P_i . Nous remarquons que cette convexité est quantifiée dans l'intervalle $[-1; 1]$, avec des valeurs négatives pour les primitives concaves et positives pour les primitives convexes.

Convexité des intersections

Comme pour les relations de tangence ou de coïncidence, la détermination de la convexité d'une intersection est spécifique aux types des primitives prises en compte. Rappelons qu'à cette étape du processus, les primitives sont délimitées par leurs sous-maillages. Nous pouvons alors déterminer si une primitive P_1 est à l'intérieur (noté **in**) ou à l'extérieur (noté **out**) d'une primitive P_2 . Pour cela, nous proposons de calculer les distances signées entre les points du sous-maillage de P_1 et la surface de P_2 . Si toutes les distances sont négatives, alors P_1 **in** P_2 . De même, P_1 **out** P_2 si les distances sont positives. Enfin, des distances négatives et positives suggèrent que P_1 passe à travers P_2 , noté P_1 **through** P_2 . Notons que ces trois relations (**in**, **out** et **through**) ne sont pas commutatives.

Ainsi, nous définissons une liste de conditions permettant de déterminer si une intersection entre deux primitives voisines P_1 et P_2 est convexe ou concave.

- Si P_1 est un plan, l'intersection entre P_1 et P_2 est :
 - convexe si P_2 **in** P_1 ;
 - concave si P_2 **out** P_1 .
- Dans tous les autres cas, l'intersection entre P_1 et P_2 est :
 - convexe si P_1 est convexe et P_2 **in** P_1 ;
 - concave si P_1 est concave et P_2 **in** P_1 ;
 - convexe si P_1 est concave et P_2 **out** P_1 ;
 - concave si P_1 est convexe et P_2 **out** P_1 ;
 - convexe si P_1 est concave et P_2 **through** P_1 ;
 - concave si P_1 est convexe et P_2 **through** P_1 .

A partir de toutes ces relations, nous pouvons construire un graphe, comme illustré dans la figure 4.3. Pour plus de clarté, nous le représentons ici en plusieurs couches.

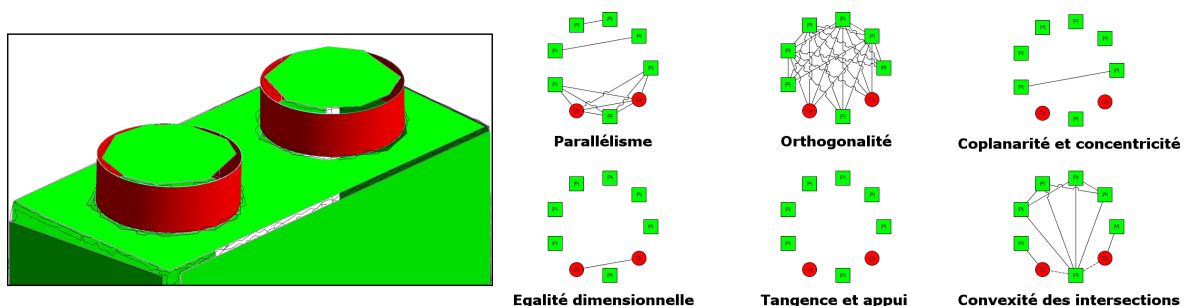


FIGURE 4.3 – Exemple de graphe de relations en plusieurs couches (plans en vert et cylindres en rouge).

4.3 Analyse globale et embellissement des primitives

Nous avons vu dans le chapitre 2 que la modélisation CAO impliquait des contraintes de construction, induites par un *système de référence*. Celui-ci contient principalement un repère orthonormé, ainsi qu'un ensemble de plans, de droites et de points de référence. La modélisation d'une pièce consiste alors à construire un ensemble de primitives respectant des relations géométriques par rapport à ces différentes références. Par exemple, un ensemble de cylindres parallèles peut être défini par une relation globale de parallélisme par rapport à un axe privilégié.

L'objectif est donc de retrouver ce système de référence à partir de l'ensemble des primitives extraites et du graphe de relations. Pour cela, nous proposons un processus en trois étapes (Fig. 4.4). Dans un premier temps, les orientations des primitives sont analysées afin d'en déduire un repère orthonormé de référence. Les paramètres d'orientation des primitives sont alors régularisés. Dans un second temps, les positions des primitives régularisées sont analysées afin d'en déduire des plans de références. Ces derniers permettent alors de régulariser à nouveau les primitives. Enfin, les dimensions des primitives sont analysées puis régularisées. Les références extraites permettent ainsi de régulariser les paramètres des primitives afin d'obtenir un modèle correspondant au mieux au modèle initial d'une pièce.

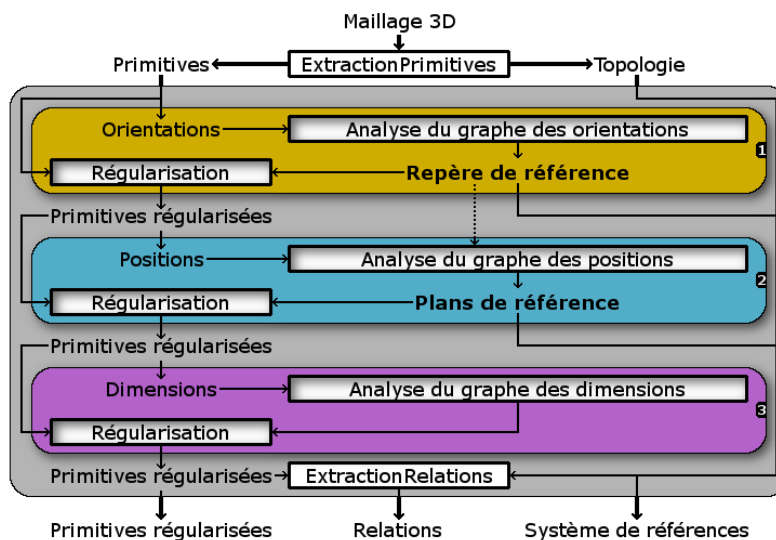


FIGURE 4.4 – Schéma global du processus d'analyse et de régularisation des primitives.

Un des points clés de notre méthode est la réalisation de trois régularisations successives. En effet, la première régularisation, à partir d'un graphe d'orientation, permet d'obtenir des primitives exactement parallèles. Cela facilite alors la détection de la relation de concentricité, par exemple, qui nécessite de calculer une distance entre deux axes. Le graphe de relations est alors calculé à nouveau pour régulariser les positions. Enfin, le graphe est à nouveau mis à jour pour régulariser les dimensions.

4.3.1 Analyse des orientations des primitives

Comme nous l'avons vu dans le chapitre 2, la modélisation CAO s'appuie généralement sur un repère orthonormé de référence afin de faciliter la construction de relations d'orientation telles que le parallélisme ou l'orthogonalité. Dans le cadre de la rétro-ingénierie, nous n'avons aucun repère *a priori*. Nous proposons alors d'estimer un repère à partir des orientations des primitives extraites, puis de régulariser ces orientations afin qu'elles correspondent au mieux à celles du modèle initial.

4.3.1.1 Reconnaissance d'un repère orthonormé de référence

Dans un objet 3D, nous pouvons intuitivement trouver différentes orientations (Fig. 4.5). Ces orientations sont majoritairement définies par les paramètres des primitives géométriques (par exemple l'axe d'un cylindre ou la normale à un plan), mais peuvent aussi être données par les arêtes de l'objet.

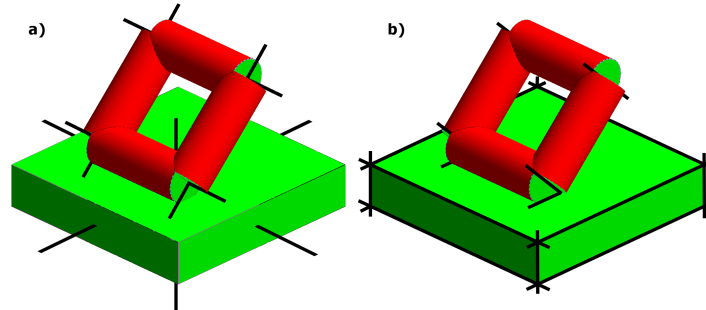


FIGURE 4.5 – Extraction des orientations : a) Paramètres des primitives, b) Arêtes de l'objet.

Comme détaillé en section 4.2, nous regroupons les orientations similaires afin d'en déduire des relations globales entre les primitives. Comme illustré dans la figure 4.6, ces relations globales peuvent être représentées sous la forme d'un graphe dans lequel :

- Un noeud *primitive* représente une orientation V_i définie par une primitive. Chaque noeud est associé à un poids ω_i , correspondant à l'aire de la primitive.
- Un noeud *groupe* représente une orientation moyenne \mathcal{O} . Chaque noeud est associé à un poids ω_k .
- Une arête reliant un noeud *primitive* et un noeud *groupe* représente l'inclusion de l'orientation associée à la primitive dans le groupe d'orientations similaires. Un ensemble d'arêtes reliées au même noeud représente implicitement la relation de parallélisme.
- Une arête est ajoutée entre deux noeuds *groupe* lorsque les orientations moyennes de ces derniers sont orthogonales à Δ_{angle} près.

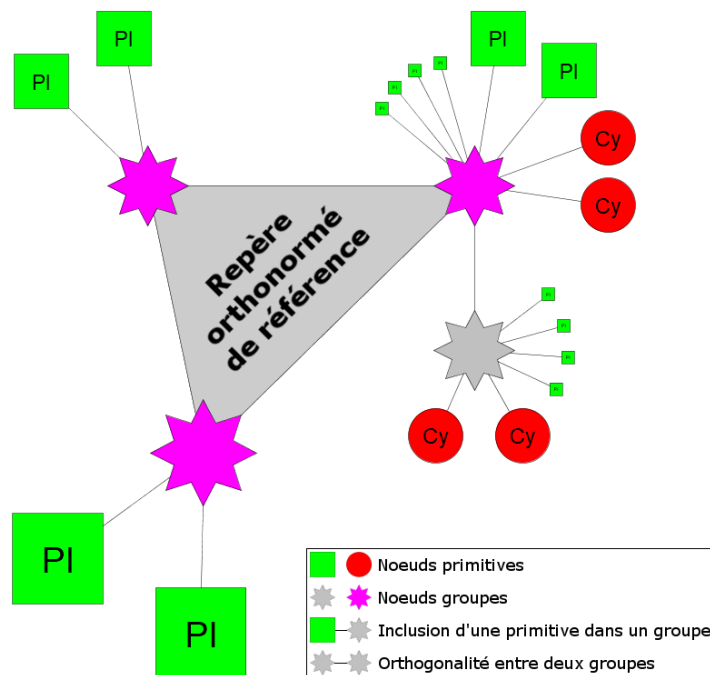


FIGURE 4.6 – Graphe des orientations extraites dans la figure 4.5.a. Les plans sont représentés par des carrés verts et les cylindres par des disques rouges. Les noeuds sont représentés de différentes tailles en fonction des poids associés.

Ce graphe permet de montrer les relations de parallélisme et d'orthogonalité entre les primitives extraites. Nous cherchons à en déduire un repère orthonormé, qui est défini par trois orientations de référence $(\vec{\mathcal{O}}_1, \vec{\mathcal{O}}_2, \vec{\mathcal{O}}_3)$ orthogonales deux à deux. Un tel repère se traduit alors par un cycle de trois noeuds *groupe*. Dans le cas où plusieurs cycles sont en concurrence, nous choisissons celui dont la somme des poids $\omega_1 + \omega_2 + \omega_3$ est la plus élevée. En effet, cela maximise la surface représentée de l'objet, puisque les poids correspondent aux aires des primitives extraites. Les trois orientations $(\vec{\mathcal{O}}_1, \vec{\mathcal{O}}_2, \vec{\mathcal{O}}_3)$ définissent alors un repère orthonormé intrinsèque à l'objet, qui peut être considéré comme le repère de référence utilisé lors de la modélisation CAO. Ainsi, nous remarquons que le repère obtenu dans la figure 4.6 correspond aux six plus grands plans, qui constituent le socle de l'objet de la figure 4.5.

S'il n'existe aucun cycle de trois noeuds *groupe* dans le graphe, nous choisissons l'arc entre deux noeuds $(\vec{\mathcal{O}}_1, \vec{\mathcal{O}}_2)$ ayant le poids maximal $\omega_1 + \omega_2$ et calculons la troisième orientation par un produit vectoriel : $\vec{\mathcal{O}}_3 = \vec{\mathcal{O}}_1 \wedge \vec{\mathcal{O}}_2$.

S'il n'existe aucun cycle ou arc, nous choisissons le noeud ayant le poids maximal comme unique axe de référence. Cela peut se produire pour les objets de révolution par exemple.

Notons en particulier que chaque axe obtenu est associé à un poids différent. Cela permet alors d'indiquer un ordre d'importance des axes, pouvant par exemple indiquer un axe d'usinage ou de démoulage de la pièce. Le repère orthonormé de référence est par la suite utilisé en tant que référence externe et absolue, comme en modélisation CAO.

4.3.1.2 Régularisation des orientations

Pour régulariser les orientations des primitives à partir du repère de référence obtenu, nous proposons d'utiliser un ensemble d'angles particuliers Ω_{angles} représentant les inclinaisons possibles des primitives par rapport aux plans du repère. Nous pouvons alors définir une orientation de référence pour chaque couple d'angles (ϕ, θ) , comme illustré dans la figure 4.7.a. Ces orientations de référence permettent alors de construire une sphère de quantification (Fig. 4.7.b).

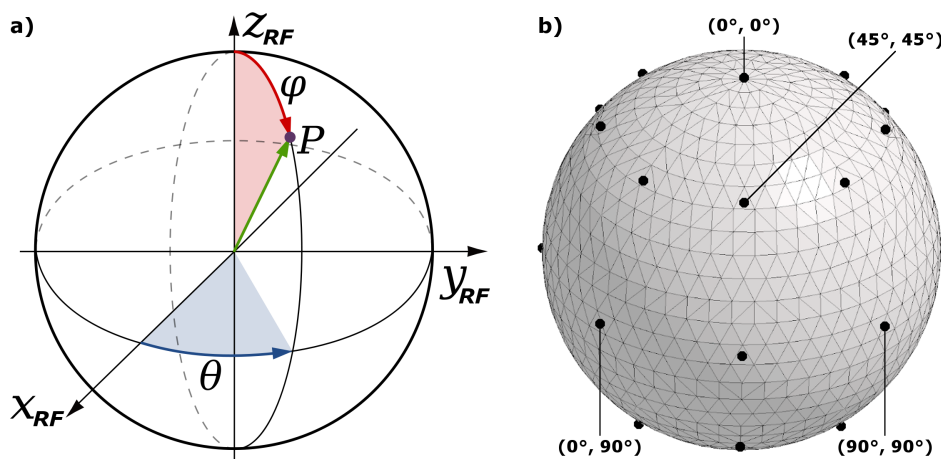


FIGURE 4.7 – Construction d'une sphère de quantification : a) Représentation d'une orientation de référence (ϕ, θ) sous forme de coordonnées sphériques, b) Sphère de quantification avec $\phi = k \times 45^\circ$ et $\theta = l \times 45^\circ$.

Enfin, l'orientation moyenne $\vec{\mathcal{O}}_i$ de chaque groupe peut être régularisée en fonction de l'orientation de référence la plus proche $\vec{\mathcal{O}}_{ref}$, si et seulement si $\vec{\mathcal{O}}_i \vec{\mathcal{O}}_{ref} \leq \Delta_{angle}$. Chaque primitive appartenant à ce groupe est alors régularisée en quantifiant son orientation \vec{V}_i avec la même orientation de référence, ce qui permet de respecter exactement les relations de parallélisme. Plus généralement, les primitives régularisées respectent des relations d'angles particuliers en fonction des angles définis par Ω_{angles} .

4.3.1.3 Optimisation contrainte des primitives

La dernière étape de la régularisation des orientations consiste à optimiser les autres paramètres des primitives. En effet, il n'est pas suffisant de modifier les paramètres d'orientation en laissant les paramètres de position et de dimensions inchangés.

La figure 4.8.a montre un tronçon de cône dont le sommet est très éloigné, ce qui est souvent le cas pour des cônes à faible demi-angle. Une régularisation, même légère, de l'orientation du cône peut alors engendrer des déviations importantes entre la primitive et son sous-maillage support (Fig. 4.8.b). Nous devons alors réaliser une optimisation contrainte des primitives régularisées. Dans notre exemple, il s'agit de recalculer le sommet et le demi-angle du cône, sans modifier son orientation déjà régularisée (Fig. 4.8.c).

Pour cela, nous proposons d'adapter l'algorithme d'optimisation présenté dans le chapitre 3 (voir section 3.3). Pour rappel, l'optimisation est généralement basée sur une descente de gradient, et consiste à estimer les meilleurs paramètres de la primitive afin de minimiser la somme quadratique des distances entre les points du sous-maillage support et la surface de la primitive. Afin de contraindre cette optimisation, nous proposons de ne pas prendre en compte les composantes du gradient liées à l'orientation.

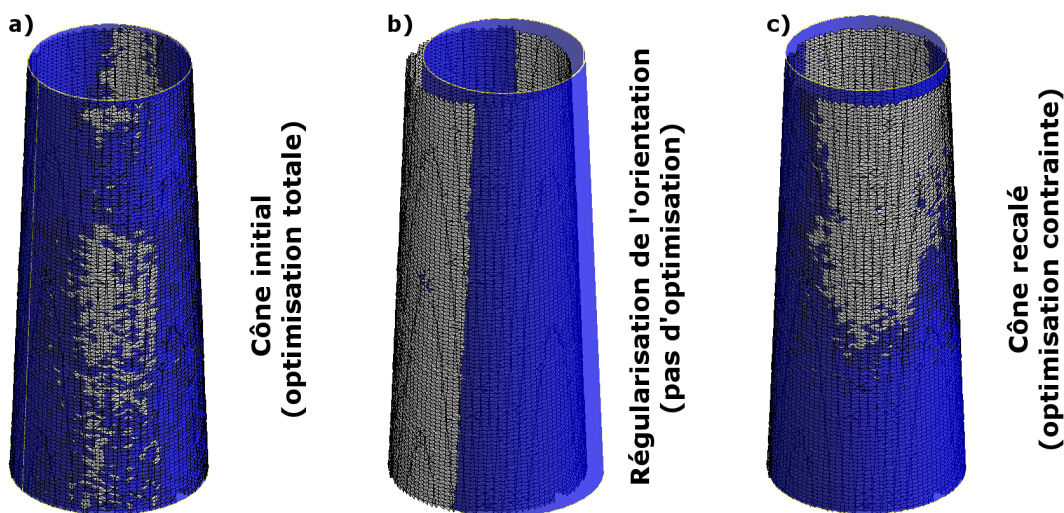


FIGURE 4.8 – Optimisation contrainte d'un cône régularisé sans modification de son orientation : a) Cône ajusté sur le sous-maillage, b) Régularisation engendrant une déviation importante, c) Cône recalé.

4.3.2 Analyse des positions des primitives

La modélisation CAO d'une pièce manufacturée s'appuie généralement sur des positionnements particuliers des primitives, généralement à l'aide d'une grille régulière. Cette grille peut alors être vue comme un ensemble de plans parallèles et orthogonaux de référence, qui facilitent ainsi la création de relations de concentricité, de coplanarité, de tangence ou d'alignement entre les primitives. Nous proposons d'estimer cette grille à partir des primitives, à travers une reconnaissance de plans de référence.

4.3.2.1 Reconnaissance de plans de référence

Dans un premier temps, nous devons retrouver les éléments potentiellement alignés. Ces derniers sont majoritairement définis par des axes et des sommets extraits des primitives. Ainsi, nous pouvons par exemple extraire les axes des cylindres et des cônes, les droites d'intersection entre les plans voisins, les sommets des cônes et les centres des sphères. Il est également possible d'étendre l'ensemble des points en calculant les intersections entre les droites obtenues. Ces éléments définissent en quelque sorte un squelette de l'objet (Fig. 4.9).

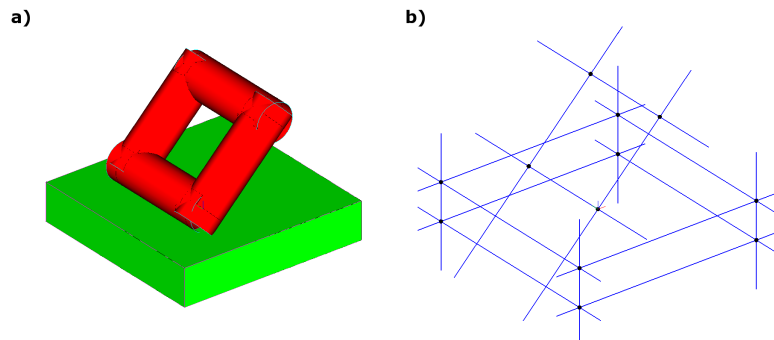


FIGURE 4.9 – Extraction du squelette d'un objet : a) Primitives, b) Axes, sommets et intersections.

Mais ces éléments, de la même manière que les primitives, sont imprécis. Dans ce cas, deux cylindres censés être concentriques peuvent fournir deux axes différents. Nous proposons alors de regrouper les droites et points similaires, comme nous l'avons fait pour les orientations en section 4.3.1. Ainsi, nous définissons deux éléments comme étant similaires si la distance euclidienne qui les sépare est inférieure à $\Delta_{distance}$. Après regroupement de ces différents éléments, nous calculons pour chaque groupe une position moyenne en pondérant chaque élément par l'aire de la primitive associée. Pour un groupe de points, un barycentre pondéré est calculé. Pour un groupe d'axes, censés être exactement parallèles après régularisation des orientations, nous proposons de projeter un point arbitrairement choisi sur chaque axe, puis de calculer le barycentre pondéré des projections avec l'équation 4.4. Implicitement, ce processus regroupe les positions identiques, permettant d'extraire des relations globales entre les primitives. Notons en particulier que le regroupement des axes ne peut être précis que s'ils sont exactement parallèles, ce qui est rendu possible par la régularisation préalable des orientations (voir section 4.3.1).

Nous pouvons alors trouver des régularités dans les positions des primitives, induisant souvent de nombreux alignements. La plupart d'entre eux correspondent à des plans ou droites de référence, ayant souvent une orientation particulière définie à partir du repère orthonormé de référence. Puisque ce repère a déjà été extrait précédemment, il est possible d'en déduire tous les plans de référence possibles, permettant d'aligner au moins deux primitives. Ainsi, nous cherchons à construire des plans formant des angles particuliers avec le repère orthonormé, définis par Ω_{angles} , et contenant au moins deux axes ou sommets moyens (Fig. 4.10). Pour cela, nous réalisons les cinq étapes suivantes.

1. Construire un plan avec une orientation de référence définie par un couple d'angles particuliers prédéfinis par Ω_{angles} (voir section 4.3.1).
2. Translater le plan à la position moyenne d'un groupe. Cela crée un nouvel alignement.
3. Retrouver les éléments moyens appartenant au plan construit :
 - les sommets séparés du plan par une distance inférieure à $\Delta_{distance}$;
 - les axes dont le point appartient au plan et ayant une orientation orthogonale au plan.
4. Répéter les étapes précédentes jusqu'à avoir traité chaque orientation de référence avec chaque groupe.
5. Conserver uniquement les plans de référence associés à plusieurs groupes, puis calculer leur position finale par une moyenne pondérée.

Enfin, les plans de référence associés à des éléments alignés sur une droite sont transformés en droites de référence. Il est également possible de considérer les groupes restants comme des éléments de référence, ou encore fusionner les plans de référence avec les primitives planes similaires, en fonction des tolérances $\Delta_{distance}$ et Δ_{angle} .

En modélisation CAO, il est fréquent de contraindre la construction d'un modèle à l'aide d'une grille 2D ou 3D régulière. Dans ce sens, nous remarquons que l'ensemble des plans de référence de la figure 4.10.a, qui ont pour orientations les axes du repère orthonormé de référence, représente une grille 3D semi-régulière. Mais notre méthode va également plus loin en proposant des plans

de référence inclinés en fonction d'angles prédéfinis (Fig. 4.10.b et 4.10.c). Ainsi, nous pouvons en déduire des relations d'alignement plus variées, qui reflètent une intention de construction de l'objet.

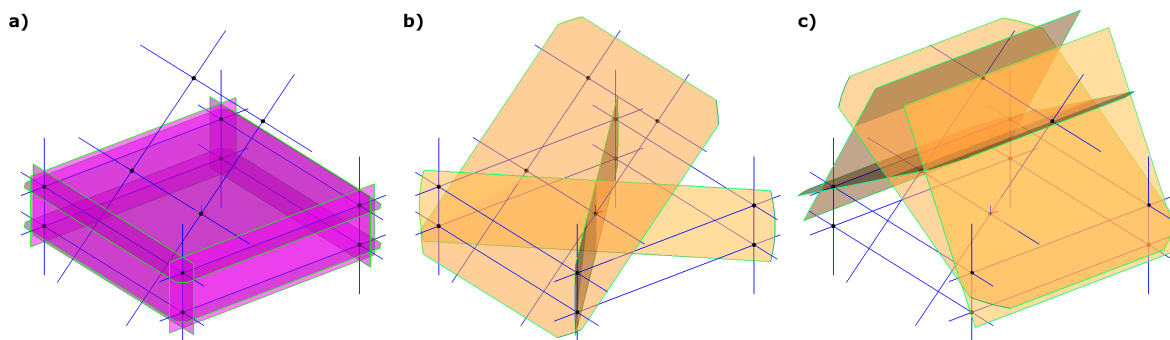


FIGURE 4.10 – Extraction des plans de référence à partir des axes et sommets, formant des angles particuliers avec les axes du repère orthonormé de référence : a) 0° , b) 45° , c) 30° .

4.3.2.2 Régularisation des positions

L'ensemble de plans de référence extraits précédemment définit un système de contraintes géométriques. Afin de satisfaire des relations d'alignement entre les primitives, il est nécessaire de résoudre ce système de contraintes. Pour cela, il existe des méthodes classiques de résolution de problèmes géométriques [77, 100]. Ces dernières fournissent alors un ensemble d'axes et sommets moyens régularisés en fonction des plans de référence. Par la suite, les positions des primitives sont régularisées en fonction des axes et sommets moyens régularisés correspondants. Enfin, comme pour la régularisation des orientations, il est nécessaire d'appliquer une optimisation contrainte pour mettre à jour les autres paramètres des primitives. Puisque les orientations et les positions ont déjà été régularisées, seules les dimensions sont optimisées. Pour cela, nous adaptons à nouveau les formules proposées par Shakarji [96].

Nous pouvons en particulier remarquer que cette régularisation permet de satisfaire de nouvelles relations entre les primitives. Par exemple, deux axes presque concentriques sont regroupés durant l'extraction des plans de référence. Ensuite, les positions des deux axes sont régularisées en fonction de l'axe moyen du groupe, les rendant exactement concentriques. De même, la régularisation des positions permet d'améliorer les intersections entre les axes et sommets moyens, qui ne s'intersectent pas exactement au départ. D'autres relations sont ainsi régularisées, comme la coplanarité ou la coïncidence (voir section 4.2).

Un autre point important intervient lorsqu'une primitive plane correspond exactement à un plan de référence après régularisation. En effet, un tel plan est probablement construit avant les autres primitives, qui peuvent ensuite être alignées sur ce plan par exemple. Nous pouvons alors en déduire un ordre de priorité entre certaines primitives, qui fait partie des informations de conception de l'objet telles que la construction d'un socle par exemple.

Cependant, l'utilisation d'un tel système de contraintes admet des limites. Premièrement, de nombreux plans de référence peuvent engendrer un système sur-contraint et donc plus difficile à résoudre (Fig. 4.11.a). Deuxièmement, de très nombreuses contraintes peuvent être générées, notamment pour les objets complexes, ce qui augmente significativement la complexité et donc les temps de calcul des méthodes de résolution. Afin d'éviter ce problème, nous devons simplifier le système de contraintes, par exemple en supprimant les plans de référence qui ne sont pas nécessaires. La principale difficulté concerne alors le choix des plans à supprimer, tout en faisant attention à ne pas créer un système sous-contraint (Fig. 4.11.b). Dans nos travaux futurs, nous aimerions quantifier la probabilité qu'un alignement puisse être utilisé lors de la conception initiale de l'objet. Ces probabilités pourraient alors permettre de guider la résolution du système. Nous pourrions aussi nous appuyer sur d'autres relations telles que la topologie, pouvant donner une priorité aux relations entre primitives voisines par exemple.

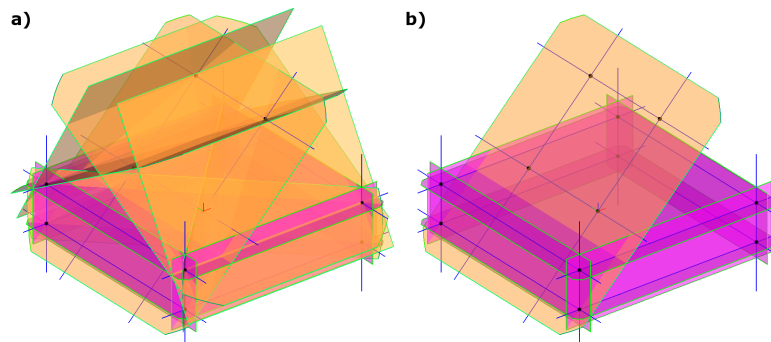


FIGURE 4.11 – Exemple d'un système de plans de référence : a) Sur-contraint, b) Sous-contraint.

4.3.3 Analyse des dimensions des primitives

Comme pour les orientations et les positions, les paramètres de dimensions des primitives sont imprécis. Ainsi, nous cherchons à régulariser ces dimensions afin de respecter exactement les relations d'égalité entre deux rayons ou deux demi-angles par exemple. Pour cela, nous nous basons principalement sur le regroupement des dimensions similaires, en différenciant leur type : rayons, demi-angles et hauteurs. Ensuite, une dimension de référence est calculée pour chaque groupe à l'aide d'une moyenne pondérée, et toutes les primitives associées sont régularisées en fonction de cette valeur.

En modélisation CAO, il est en effet fréquent de construire une pièce avec des dimensions régulières, comme par exemple des diamètres et profondeurs de perçage. Ces dimensions peuvent également correspondre à des normes, qui sont souvent présentées sous forme de tables métriques et qui représentent des règles métier (voir tableau 4.1).

Size - Nominal Diameter (mm)	Pitch (mm)	Clearance Drill (mm)	Tap Drill (mm)
M 1.60	0.35	1.80	1.25
M 2.00	0.40	2.40	1.60
M 2.50	0.45	2.90	2.00
M 3.00	0.50	3.40	2.50
M 3.50	0.60	3.90	2.90
M 4.00	0.70	4.50	3.30
M 5.00	0.80	5.50	4.20
M 6.00	1.00	6.60	5.00
M 8.00	1.25	9.00	6.80
M 10.00	1.50	12.00	8.50

TABLEAU 4.1 – Exemple de table métrique pour les taraudages, issue de la norme ISO 724 :1993.

Il est alors possible de quantifier chaque dimension moyenne en fonction de la valeur normalisée la plus proche. Si de telles tables ne sont pas définies au préalable, nous ne pouvons pas garantir l'exactitude des dimensions régularisées.

4.3.4 Résultats expérimentaux

Nous avons appliqué notre méthode d'analyse et régularisation des primitives à plusieurs maillages 3D issus de numérisation. Tous ces résultats ont été obtenus avec $\Delta_{distance}$ égal à la longueur moyenne des arêtes du maillage, $\Delta_{angle} = 1^\circ$ et $\Omega_{angles} = \{0^\circ, 30^\circ, 45^\circ, 90^\circ\}$. Tout d'abord, la section 4.3.4.1 présente en détails les résultats obtenus à chaque étape sur le modèle *Lego* (Fig. 4.12). La section 4.3.4.2 décrit ensuite quelques résultats sur d'autres objets plus complexes.

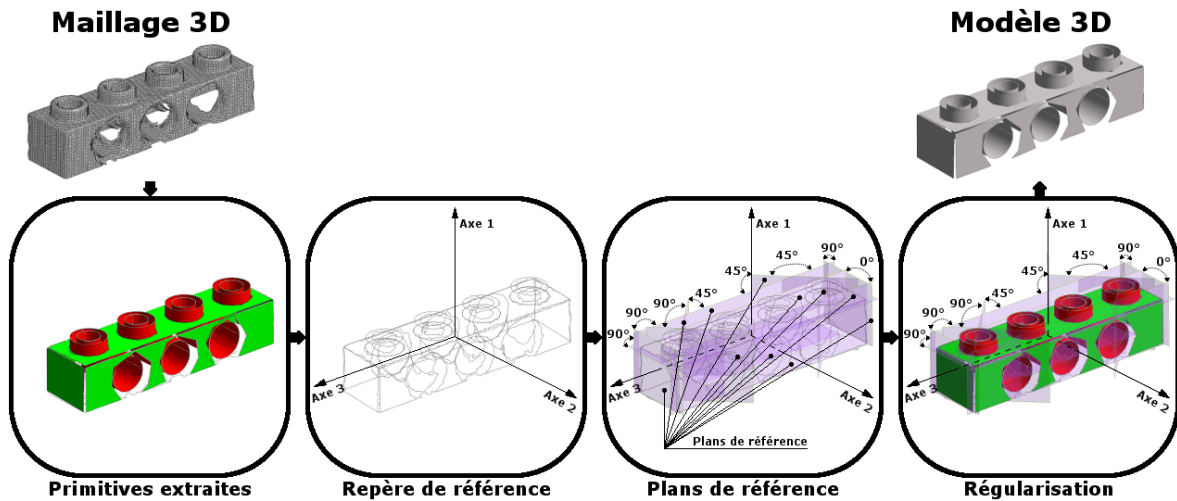


FIGURE 4.12 – Présentation du processus général d'analyse et embellissement des primitives.

4.3.4.1 Etude détaillée sur le modèle *Lego*

Le maillage *Lego* représente un parallélépipède rectangle, surmonté de quatre plots alignés et avec trois perçages latéraux, alignés eux aussi (Fig. 4.13). Comme nous pouvons le constater, ce maillage est bruité. Après extraction des primitives, nous calculons un graphe de relations.

Lego :

12 500 points
24 371 triangles

31,82 mm de long
7,98 mm de large
8,59 mm de haut

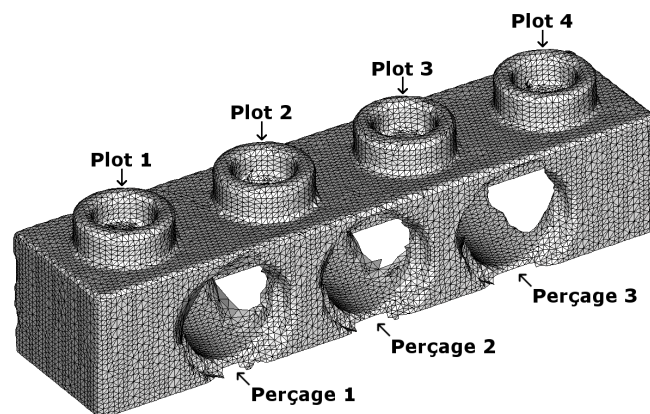


FIGURE 4.13 – Présentation du maillage *Lego*.

La figure 4.14 illustre le sous-graphe correspondant aux relations d'adjacence. La topologie calculée est correcte et correspond à celle du modèle initial. Il est à noter que les sous-maillages restants, ici correspondant à des demi-tores, sont bien reconnus.

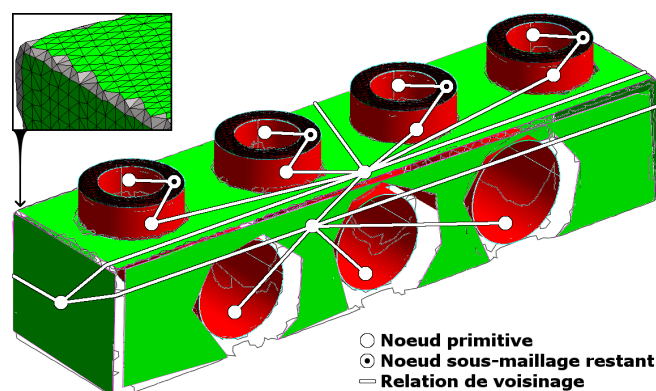


FIGURE 4.14 – Relations d'adjacence entre les primitives extraites de *Lego*.

La figure 4.15 montre le sous-graphe correspondant aux orientations, qui contient un cycle de trois noeuds groupes. Nous pouvons voir que le repère orthonormé de référence associé à ce cycle est adapté à l'objet. Notons en particulier que l'axe principal correspond aux axes des quatre plots. Le second axe correspond quant à lui aux axes des trois perçages latéraux de la brique. Le troisième axe permet alors de compléter le repère orthonormé.

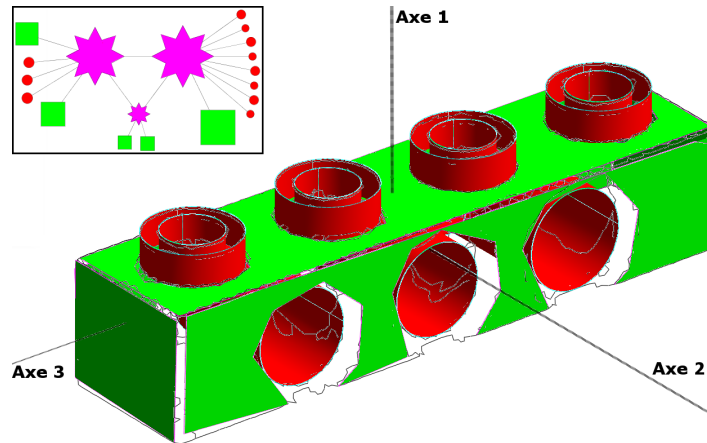


FIGURE 4.15 – Repère orthonormé de référence calculé à partir des primitives de *Lego*.

Dans la figure 4.16, nous présentons des relations d'orientation entre les primitives de *Lego*, en particulier le parallélisme et l'orthogonalité. Pour cela, nous analysons les angles formés entre l'orientation d'une primitive de référence et les orientations des autres primitives. Les paramètres des primitives extraites étant imprécis, nous trouvons des angles entre les orientations parallèles entre 0° et 3° , et entre 87° et 90° pour les orientations orthogonales (Fig. 4.16.a). Notons que ces angles sont toujours ramenés à l'intervalle $[0;90]$ puisqu'ils sont calculés entre des droites et non des vecteurs. Après le processus de régularisation des orientations, nous obtenons des angles égaux à 0° et 90° (Fig. 4.16.b). Le point clé de notre approche est de réaliser une régularisation globale pour laquelle chaque primitive est indépendante. Au final, les cylindres des quatre plots et des trois perçages deviennent exactement parallèles ou orthogonaux.

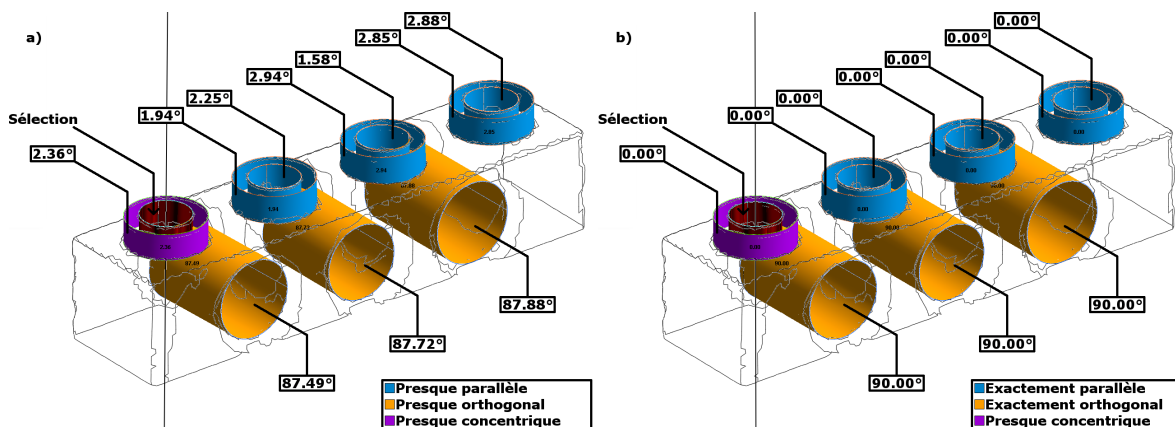


FIGURE 4.16 – Angles entre l'orientation de la primitive sélectionnée et celles des autres primitives de *Lego* : a) Avant et b) Après régularisation.

A partir des primitives régularisées, nous pouvons calculer de manière plus précise la distance entre ces axes deux à deux, par exemple pour déterminer la concentricité des cylindres extérieurs et intérieurs des plots. Cependant, comme ces primitives sont imprécises, aucune paire d'axes n'est exactement concentrique (Fig. 4.16.b). Il est alors nécessaire d'appliquer une régularisation des positions.

Pour cela, nous calculons des plans de référence à partir des axes des cylindres et des droites d'intersection entre les plans voisins. Comme nous avons déjà appliqué notre régularisation des

orientations, toutes ces droites sont exactement parallèles ou orthogonales. Pour autant, ces droites ne s'intersectent pas exactement. Dans ce cas, le point d'intersection est approximé par un barycentre pondéré. Ensuite, nous pouvons extraire des plans de référence contenant plusieurs droites ou points. La figure 4.17 montre les plans de référence extraits à partir de *Lego*, ainsi que les primitives correspondantes après régularisation des positions.

Dans la figure 4.17.a, nous pouvons voir deux plans de référence parallèles à l'axe principal du repère de référence. En particulier, l'un d'eux contient les axes des trois perçages latéraux. Dans la figure 4.17.b, nous trouvons trois plans de référence parallèles au second axe du repère de référence. En particulier, l'un d'eux contient les axes des quatre plots. La figure 4.17.c montre deux plans de référence supplémentaires, parallèles au troisième axe du repère de référence. Dans la figure 4.17.d, nous présentons d'autres plans de référence, inclinés cette fois à 45° par rapport au repère orthonormé de référence. Ces quatre plans de référence suggèrent que les plots sont centrés par rapport aux faces latérales de la brique. Notre ensemble de plans de référence ne contient ici aucune redondance. Après régularisation des positions, nous obtenons des primitives exactement alignées, ce qui correspond à la conception originale de la brique. Notons également que les cylindres des quatre plots sont maintenant exactement concentriques.

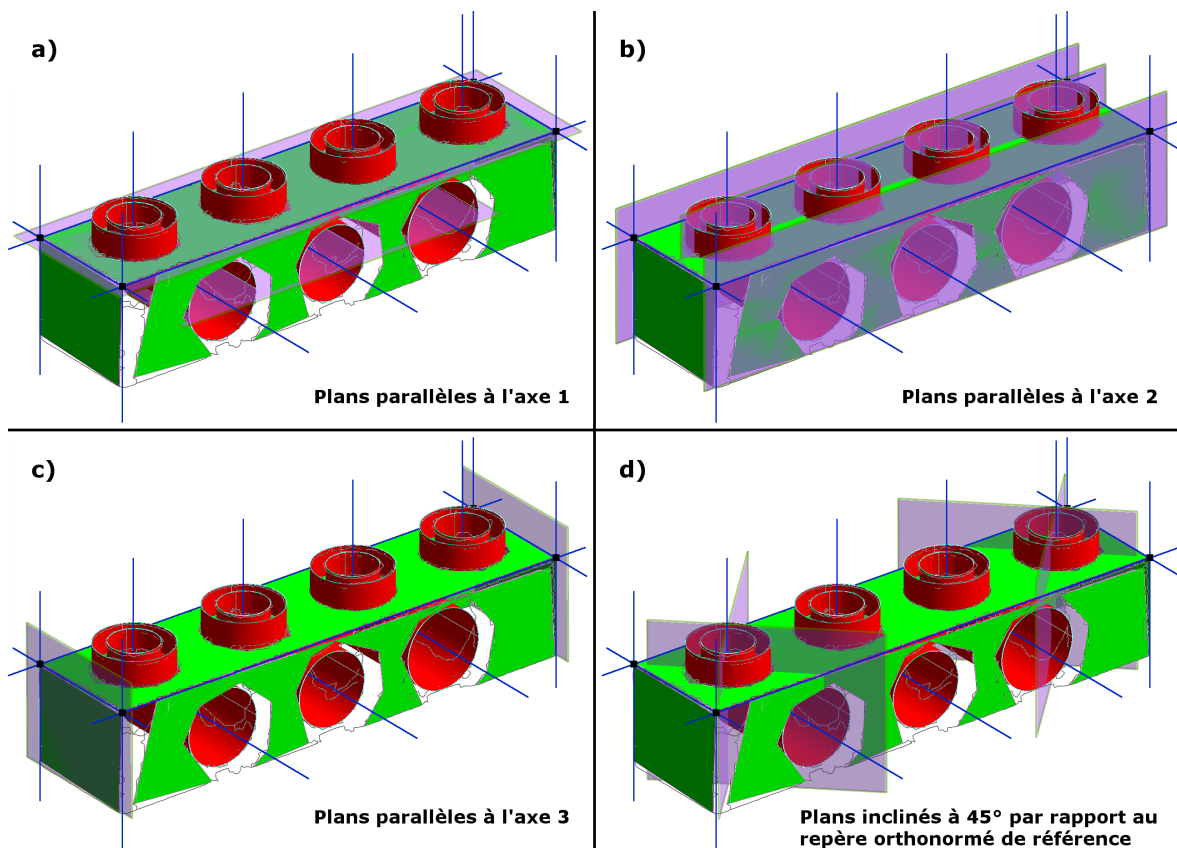


FIGURE 4.17 – Plans de référence extraits des primitives de *Lego* : a-b-c) Parallèles à un axe du repère orthonormé de référence (Fig. 4.15), ou d) Inclinés à 45°.

Après cette seconde étape de régularisation, nous analysons les dimensions des primitives, correspondant ici aux rayons et hauteurs des cylindres de *Lego*. Ainsi, la figure 4.18 illustre le sous-graphe correspondant aux relations d'égalité dimensionnelle. Ces dernières semblent correctes vis à vis du modèle. Nous représentons ainsi les primitives ayant des dimensions similaires avec la même couleur. En particulier, nous remarquons que 7 cylindres ont des rayons proches de 2.4. Mais puisque leurs hauteurs diffèrent, nous construisons deux groupes distincts, respectivement de 3 et 4 cylindres. Afin de régulariser les dimensions des cylindres, nous recherchons les valeurs les plus proches dans une table métrique prédéfinie (à droite de la figure 4.18). Cela permet de garantir la fiabilité de notre régularisation vis-à-vis du processus de conception de la pièce originale.

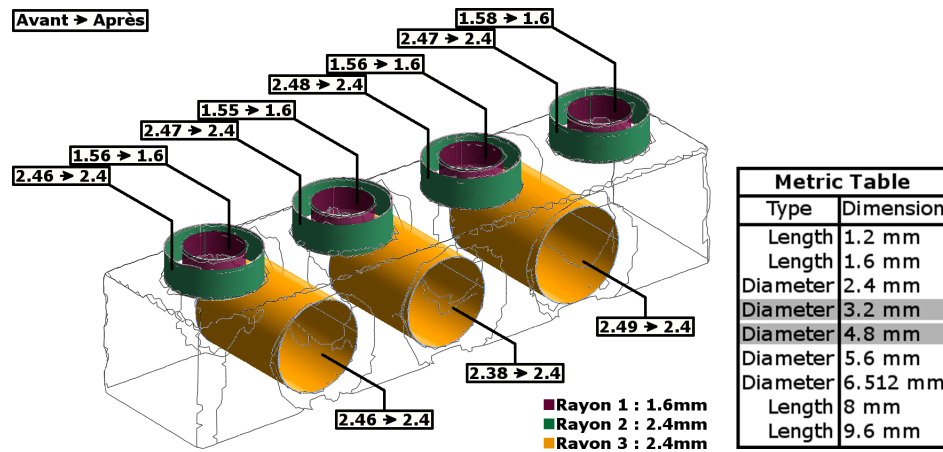


FIGURE 4.18 – Régularisation des dimensions de Lego en fonction d'une table métrique. Cette table est une conversion en millimètres des dimensions standards des briques LEGO®, originellement en pouces.

Après les trois étapes de régularisation (orientations, positions et dimensions), le modèle reconstruit semble correspondre au modèle original.

4.3.4.2 Résultats obtenus sur des modèles manufacturés complexes

Dans cette section, nous montrons des résultats de notre approche d'analyse et régularisation des primitives extraites à partir de plusieurs maillages complexes et issus de différents scanners.

1. Scanner à lumière structurée :
 - a) *Block* : 91 primitives (plans et cylindres) ;
 - b) *Moldy* : 35 primitives (plans et cylindres) ;
 - c) *Aerospace* : 105 primitives (plans, cylindres et cônes).
2. Scanner inconnu :
 - d) *Watertight* : 341 primitives (plans, cylindres et cônes) ;
 - e) *Gear* : 370 primitives (plans, cylindres et cônes).

La figure 4.19 montre des repères orthonormés de référence calculés à partir de ces cinq objets. Nous obtenons alors un repère correct pour chaque objet, qui semble correspondre à celui utilisé lors de la modélisation. En effet, les repères orthonormés de référence de *Block*, *Moldy* et *Aerospace* correspondent principalement aux socles des objets. De même, le repère de référence obtenu à partir de *Watertight* correspond au tronc principal et aux perçages latéraux. Enfin, celui de *Gear* respecte le perçage central ainsi que les deux détrompeurs de la partie supérieure.

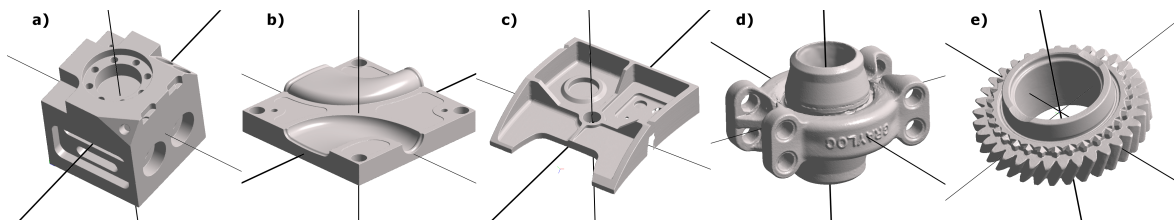


FIGURE 4.19 – Repères orthonormés de référence : a) *Block*, b) *Moldy*, c) *Aerospace*, d) *Watertight* et e) *Gear*.

Puisque ces maillages 3D issus de numérisation sont bruités, les orientations des primitives extraites sont imprécises. En effet, même le plus précis de nos modèles, à savoir *Block*, ne contient aucune relation exacte de parallélisme ou d'orthogonalité. Comme illustré dans la figure 4.20.a, les primitives extraites forment des angles situés dans l'intervalle [0.35;0.41] lorsque considérées comme parallèles et dans l'intervalle [89.61;89.97] lorsque considérées comme orthogonales. Nous pouvons voir dans la figure 4.20.b que toutes les primitives deviennent exactement parallèles ou orthogonales après notre régularisation globale.

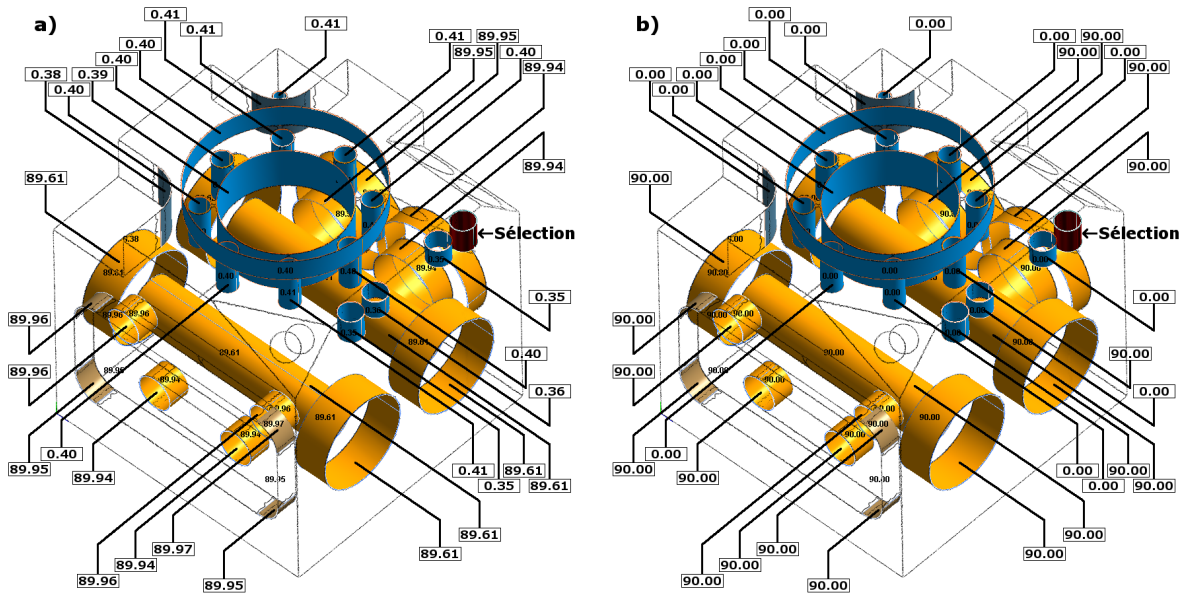


FIGURE 4.20 – Angles formés par les orientations de *Block* : a) Avant et b) Après régularisation.

Nous avons réalisé la même analyse sur deux modèles moins précis : *Aerospace* et *Watertight*. Comme illustré dans la figure 4.21.a, les orientations extraites à partir de *Aerospace* sont moins précises que pour *Block*, et peuvent dévier de plus de 1°. Après notre régularisation globale, les primitives deviennent toutes exactement parallèles ou exactement orthogonale (Fig. 4.21.b).

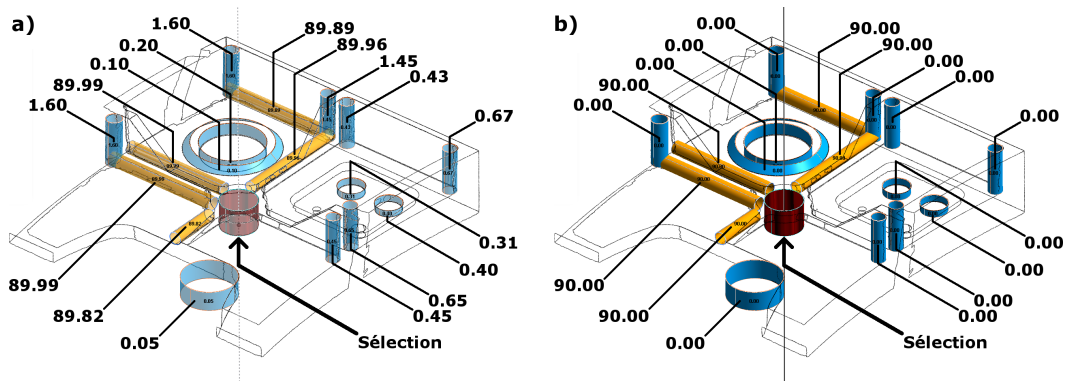


FIGURE 4.21 – Angles formés par les orientations de *Aerospace* : a) Avant et b) Après régularisation.

Comme nous pouvons voir dans la figure 4.22.a, les orientations extraites de *Watertight* sont encore moins précises et peuvent dévier de presque 2° pour certains couples de primitives. Notre processus d'analyse et régularisation des orientations permet alors d'obtenir des primitives exactement parallèles et orthogonales, comme illustré dans la figure 4.22.b.

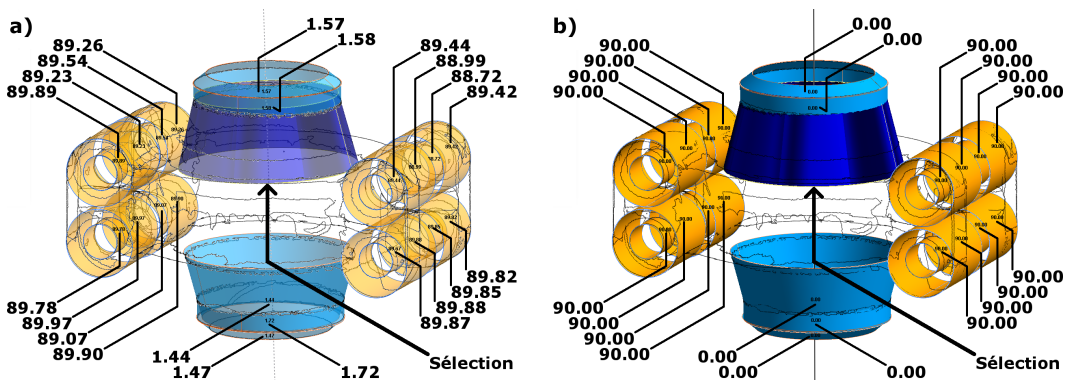


FIGURE 4.22 – Angles formés par les orientations de *Watertight* : a) Avant et b) Après régularisation.

La figure 4.23.a montre des ensembles de cylindres et cônes détectés comme concentriques, après régularisation des orientations. Nous pouvons voir que ces primitives ne sont pas exactement concentriques, ce qui se traduit par des distances non nulles entre chaque paire d'axes. Après notre seconde étape de régularisation, nous obtenons des groupes de primitives exactement concentriques (Fig. 4.23.b).

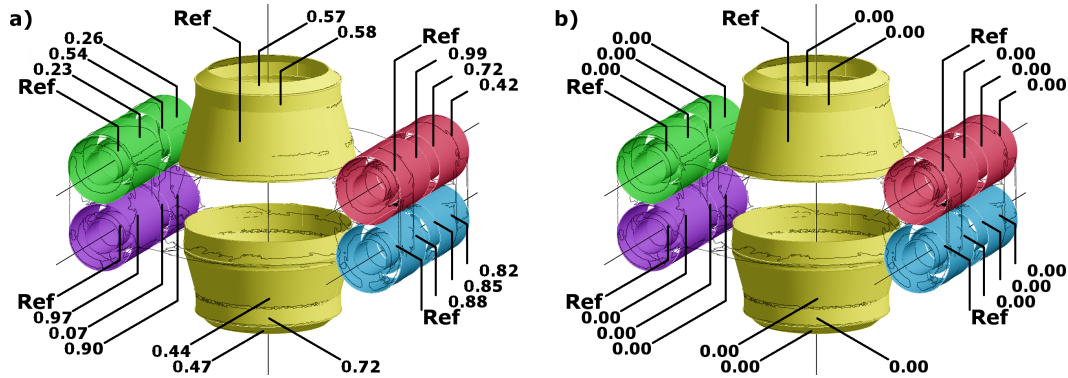


FIGURE 4.23 – Distances entre les axes concentriques de *Watertight* : a) Avant et b) Après régularisation. Chaque groupe de primitives concentriques est représenté avec une couleur différente.

La figure 4.24 montre des relations positionnelles et dimensionnelles entre les primitives de *Block*. Nous pouvons voir à travers les figures 4.24.b, 4.24.c et 4.24.d que notre méthode extrait correctement les groupes de cylindres ayant des rayons similaires. Dans la figure 4.24.e, nous présentons les relations de coplanarité entre les primitives planes. Nous remarquons en particulier que les plans coplanaires ont souvent la même forme, ce qui peut suggérer que ceux-ci sont liés. La figure 4.24.f montre les relations de concentricité entre les cylindres. Les relations de concentricité obtenues semblent correspondre à la conception initiale. Dans la figure 4.24.g, nous pouvons voir que l'objet contient également des relations de tangence, qui correspondent principalement à des congés. Ces relations permettent alors de positionner les cylindres en fonction des plans. Enfin, la figure 4.24.h montre des relations de coïncidence entre un plan et quatre demi-cylindres. En effet, l'axe d'un demi-cylindre appartient au plan de coupe de ce dernier, correspondant ici à un plan latéral de la pièce. Comme dans le cas de la tangence, cette relation permet de définir un positionnement relatif des primitives.

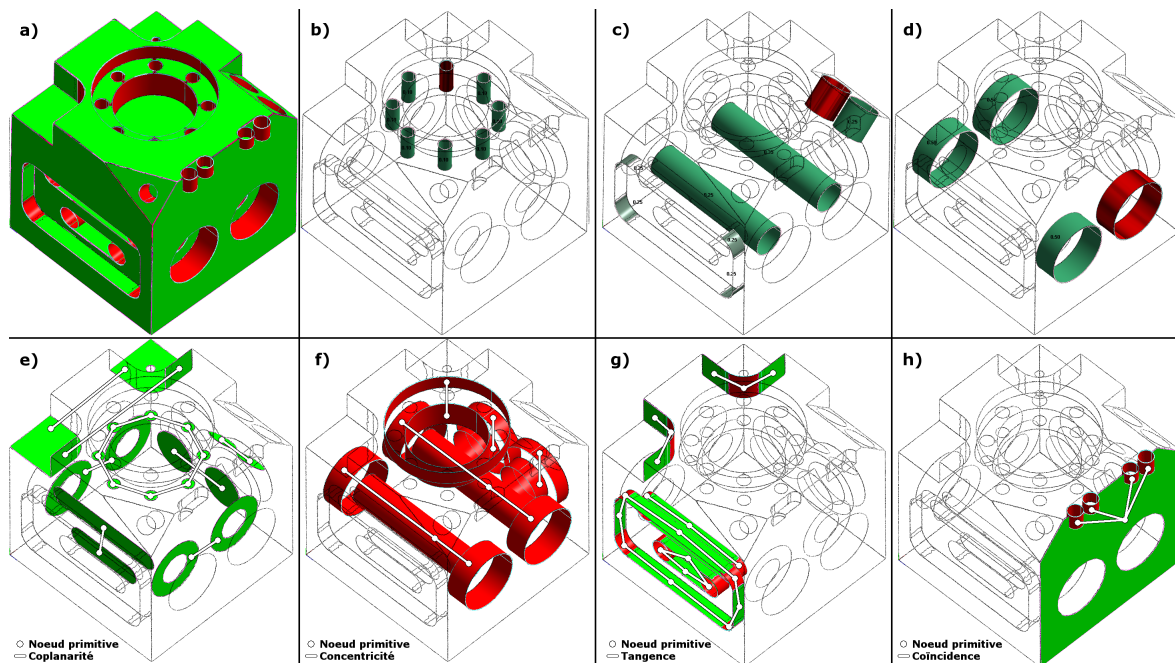


FIGURE 4.24 – Analyse positionnelle et dimensionnelle de *Block* : a) Primitives, b) c) d) Egalité dimensionnelle (rayons uniquement), e) Coplanarité, f) Concentricité, g) Tangence et h) Coïncidence.

4.4 Reconnaissance de motifs particuliers

En CAO, il est fréquent de construire des motifs particuliers de primitives selon certaines règles métier (voir chapitre 2). Dans ce chapitre, nous cherchons à reconnaître des motifs à partir de la combinaison des primitives après régularisation. Dans un premier temps, nous nous concentrons sur la détection de sous-ensembles de primitives voisines qui ont une signification d'un point de vue CAO, appelés *entités*. Dans un second temps, nous proposons de détecter des répétitions d'entités similaires. Enfin, nous proposons d'utiliser cette analyse structurelle d'une pièce manufacturée pour améliorer la régularisation géométrique des primitives.

4.4.1 Entités géométriques

En mécanique, chaque pièce fabriquée a généralement une fonction spécifique. En analysant ces dernières de manière locale, nous nous rendons également compte que certaines sous-parties d'une pièce ont elles-mêmes une fonction bien précise. Nous appelons *entité* le sous-ensemble de primitives géométriques qui composent une telle sous-partie.

Ainsi, ces entités peuvent correspondre à des fonctions mécaniques ou à des procédés de fabrication spécifiques, qui permettent de définir des relations entre les primitives géométriques qui les composent. Dans la suite, nous nous concentrons en particulier sur deux exemples d'entités : le trou lamé ou le trou fraisé (Fig. 4.25), impliquant des relations de parallélisme et de concentricité. Dans le cas d'un maillage issu de numérisation, nous ne pouvons *a priori* pas savoir s'il s'agit d'un assemblage ou d'une partie de celui-ci. Nous cherchons donc uniquement à reconnaître la structure géométrique, sans inférer de fonction mécanique ou procédé de fabrication associé.

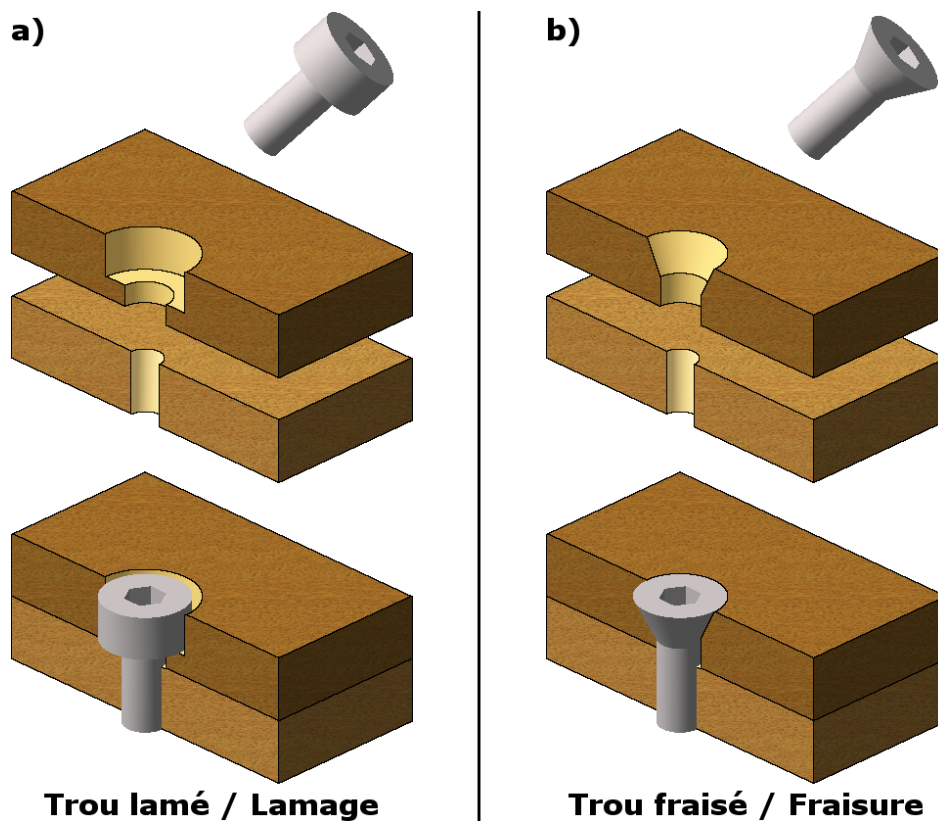


FIGURE 4.25 – Entités recherchées : a) Trou lamé, b) Trou fraisé.

Pour reconnaître ces entités, nous proposons de nous inspirer des méthodes basées sur des systèmes de règles, présentées dans le chapitre 2. Les règles définies ici correspondent aux relations présentes dans notre graphe de relations. La reconnaissance d'une entité correspond alors à l'extraction d'un sous-graphe respectant un ensemble de règles spécifiques à chaque type d'entité.

Un trou lamé, ou lamage, est un type spécifique de perçage (Fig. 4.25.a). Il a pour fonction principale d'assurer la portée de la tête d'une vis, d'un écrou ou d'une rondelle. Le lamage est fabriqué à l'aide d'un *chambrage*, permettant d'élargir la partie supérieure du trou dans le but de dissimuler la tête d'une vis ou un écrou, de manière à ce qu'ils ne dépassent pas de la surface initiale. Il est ainsi constitué d'un premier cylindre accueillant le pas de vis, d'un second cylindre plus large que le premier correspondant à la *chambre*, et enfin d'un plan sur lequel peut s'appuyer la tête d'une vis, un écrou ou une rondelle. Ainsi, nous pouvons définir un ensemble de règles permettant de reconnaître la géométrie d'un trou lamé à partir du graphe relationnel (Fig. 4.26.a) :

- le plan est voisin des deux cylindres ;
- les orientations des trois primitives sont parallèles ;
- les deux cylindres sont concaves et concentriques ;
- les deux cylindres ont des rayons différents ;
- l'intersection entre le plan et le petit cylindre est convexe ;
- l'intersection entre le plan et le gros cylindre est concave.

Un trou lamé est paramétré par une position, une orientation, deux rayons, une profondeur de perçage et une profondeur de chambrage. Sa position est alors définie comme l'intersection du plan et de l'axe commun aux cylindres.

Un trou fraisé, ou fraisure, est un autre type de perçage (Fig. 4.25.b). Sa fonction est de dissimuler une vis à tête cônique. La fraisure est fabriquée à l'aide d'un *fraisage*, permettant de remplacer l'arête débouchante du perçage par un cône. Elle est ainsi constituée d'un cylindre accueillant le pas de vis, et d'un cône sur lequel peut s'appuyer la tête d'une vis. Il est également à noter que les fraisures sont généralement construites avec des angles particuliers. Nous définissons alors un second ensemble de règles permettant de reconnaître la géométrie d'un trou fraisé à l'aide du graphe (Fig. 4.26.b) :

- le cône est voisin du cylindre ;
- le cône et le cylindre sont concaves et concentriques ;
- l'intersection entre le cône et le cylindre est convexe.

Un trou fraisé est paramétré par une position, une orientation, un rayon, un angle, une profondeur de perçage et une profondeur de fraisage. Sa position est alors définie comme le centre du cercle d'intersection entre le cône et le cylindre.

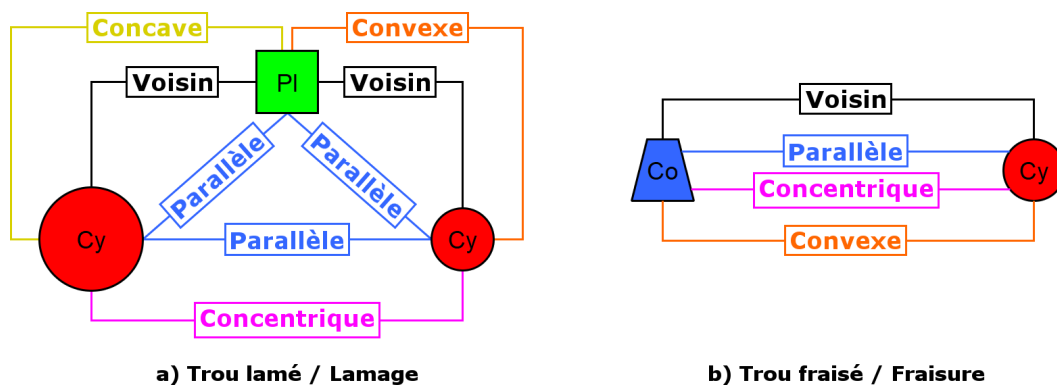


FIGURE 4.26 – Sous-graphes relationnels associés aux entités : a) Trou lamé, b) Trou fraisé.

Ainsi, chaque ensemble de règles permet de construire un sous-graphe caractéristique d'une entité. La reconnaissance de ces entités est alors réalisée en recherchant des sous-graphes caractéristiques dans le graphe de relations. Dans la suite de ce manuscrit, un cylindre concave et isolé définit une troisième entité, à savoir un perçage simple.

4.4.2 Répétitions de géométrie

En mécanique, il est fréquent de répéter des entités selon certains motifs. En effet, aligner des perçages de fixation le long d'une droite en les espaçant de manière régulière peut par exemple améliorer la robustesse globale du maintien d'une pièce, tout en accélérant la production et en réduisant les coûts de fabrication. Bien entendu, ces répétitions peuvent concerner des sous-parties diverses d'une pièce, pour des raisons économiques ou esthétiques. Ces dernières ont donc une valeur sémantique, au delà de la géométrie sous-jacente.

Nous cherchons à reconnaître des répétitions à partir de l'ensemble des entités. Ainsi, une répétition est définie comme un ensemble de plusieurs entités similaires, espacées de manière régulière et formant un motif prédéfini tel qu'une droite ou un cercle. Reconnaître des répétitions permet alors de déterminer des positions relatives entre les entités et ainsi régulariser leurs paramètres. Contrairement aux entités présentées précédemment, les répétitions peuvent s'étendre sur toute la pièce. En effet, les entités regroupées au sein d'une répétition ne sont pas nécessairement adjacentes.

4.4.2.1 Similarité entre les entités

Comme expliqué précédemment, une répétition regroupe des entités similaires. Or, dans le cadre de nos travaux, les primitives constituant ces entités sont imprécises. Il est donc nécessaire de définir au préalable cette similarité. Pour cela, nous proposons de comparer les paramètres de ces dernières, en fonction des tolérances prédéfinies (voir section 4.2). Ainsi, nous définissons deux types de similarité : partielle et complète. La similarité partielle consiste à comparer le type et les dimensions des entités. Par exemple, deux trous lamés sont partiellement similaires si leurs rayons et hauteurs sont similaires selon $\Delta_{distance}$. La similarité complète consiste, en plus de la similarité partielle, à comparer les orientations selon Δ_{angle} . Cela permet par exemple de ne regrouper que des entités parallèles. Il est à noter que les règles de similarité partielle peuvent être adaptées, en ajoutant ou supprimant des conditions, ce qui rend la méthode plus générique.

Cette analyse de similarité compare tous les paramètres des entités excepté leur position. En effet, celle-ci peut être définie de manière relative lorsqu'une entité appartient à une répétition. Nous pouvons également remarquer que la régularisation préalable des orientations permet d'améliorer la détection de similarité complète.

4.4.2.2 Détection des alignements

Les répétitions linéaires, c'est-à-dire le long d'une droite, sont les plus fréquemment utilisées en modélisation CAO. Reconnaître ces dernières consiste alors, dans un premier temps, à détecter les entités alignées. Pour cela, nous proposons de réutiliser les mêmes concepts que ceux permettant d'extraire les plans de référence (voir section 4.3.2). Ainsi, la première étape consiste à extraire un ensemble d'axes et sommets, mais cette fois à partir des entités (Fig 4.27).

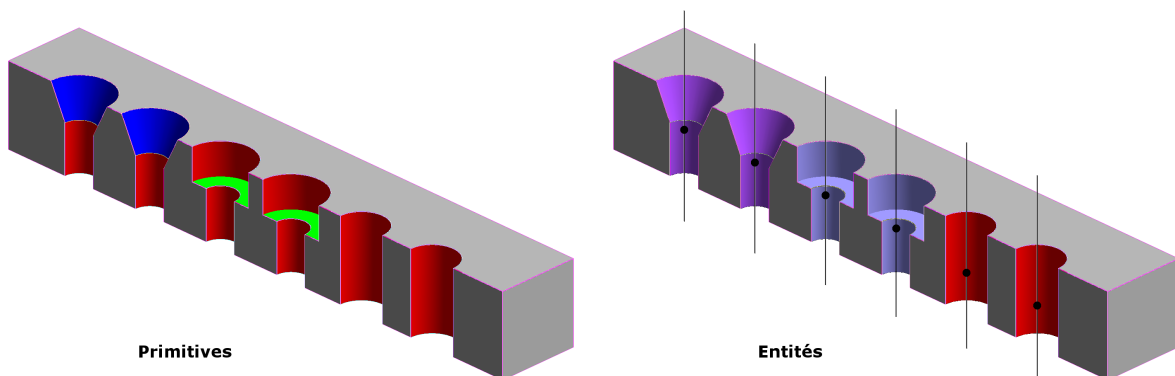


FIGURE 4.27 – Extraction des axes et sommets à partir des entités. Chaque axe ou sommet est alors considéré comme commun à toutes les primitives de l'entité correspondante.

A partir de ces axes et sommets, il est possible d'extraire des alignements selon plusieurs directions. En modélisation CAO, les répétitions se construisent généralement à partir de motifs 2D, dessinés dans un plan. En partant de ce constat, nous proposons de détecter de tels motifs en fonction des orientations de référence préalablement construites avec le repère orthonormé de référence (voir section 4.3.1). Il est à noter que dans le cas d'alignement de plusieurs axes parallèles, seul le plan orthogonal aux axes est utilisé. La figure 4.28 illustre les alignements recherchés à partir des entités de la figure 4.27.

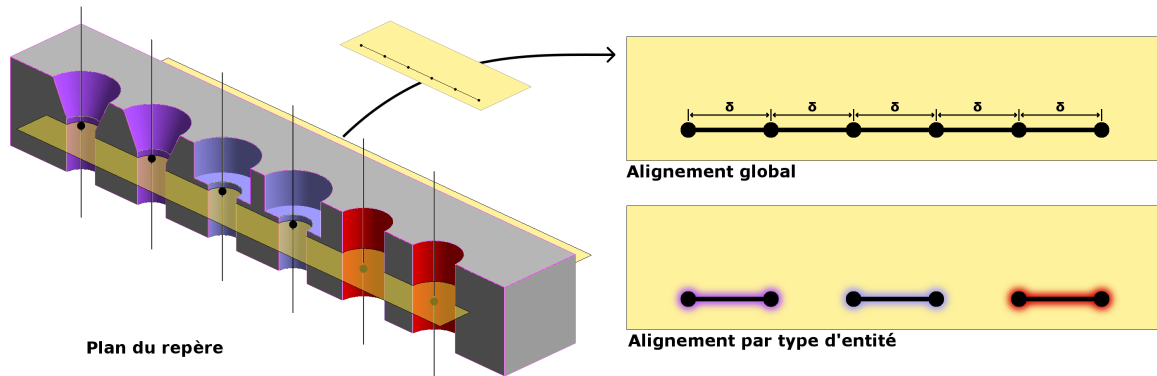


FIGURE 4.28 – Alignement des axes et sommets formant des motifs 2D dans un plan.

La détection des alignements proposée consiste à réaliser les sept étapes suivantes.

1. Construire un plan \mathcal{P} avec une orientation de référence (voir section 4.3.1).
2. Projeter les positions $\{p_i\}$ des entités sur \mathcal{P} , donnant les positions $\{p'_i\}$.
3. Construire une droite \mathcal{D} avec une orientation de référence orthogonale à celle de \mathcal{P} .
4. Translater \mathcal{D} à une position p'_i . Cela crée un nouvel alignement \mathcal{A} .
5. Ajouter à \mathcal{A} les positions $\{p'_j\}$ séparées de \mathcal{D} par une distance $\delta \leq \Delta_{distance}$.
6. Répéter ces étapes jusqu'à avoir traité tous les plans, toutes les droites et toutes les entités.
7. Conserver uniquement les alignements de plusieurs entités, puis calculer la droite finale par une moyenne pondérée avec l'aire couverte par chaque entité.

Chaque alignement est alors défini par une position, une orientation de plan, une orientation de droite et une liste d'entités. L'algorithme permet ici de retrouver des alignements globaux, comme lors de l'extraction des plans de référence (voir section 4.3.2). Pour retrouver des alignements ne regroupant que des entités similaires, il suffit de contraindre la cinquième étape pour ne récupérer qu'un certain type de position.

4.4.2.3 Reconnaissance des répétitions

Comme expliqué précédemment, une répétition est un alignement particulier regroupant des entités similaires et régulièrement espacées. En modélisation CAO, les répétitions sont généralement construites à partir de cinq paramètres :

- un plan de référence;
- un motif 2D appartenant au plan, pouvant être une droite ou un cercle par exemple;
- une entité de référence à répéter;
- le nombre d'exemplaires de l'entité que l'on souhaite;
- la distance séparant deux entités successives.

Pour généraliser la méthode, nous proposons de rechercher des combinaisons de répétitions permettant de reconstruire des dispositions complexes d'entités. Par exemple, une disposition en forme de rectangle peut être représentée soit comme une combinaison de quatre répétitions, soit

comme une répétition de deux sous-répétitions (Fig. 4.29). Il semble pour cela nécessaire de détecter les répétitions de manière récursive. Nous définissons alors un nouveau paramètre, le *degré*, qui représente la profondeur de récursion nécessaire pour construire une répétition.

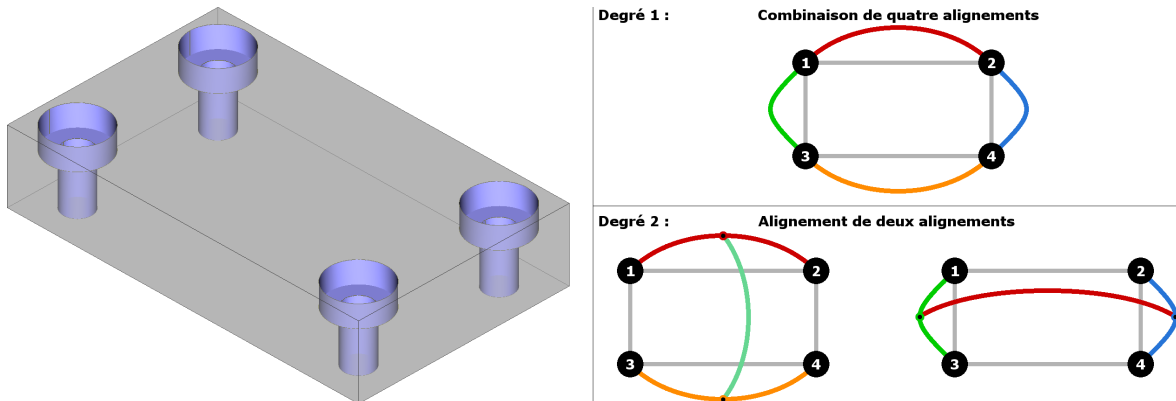


FIGURE 4.29 – Interprétations possibles d'un motif en forme de rectangle.

Nous proposons ainsi de définir une répétition p_i^d , où d correspond au degré de la répétition, comme une relation R^d entre des sous-répétitions alignées et de degré immédiatement inférieur :

$$p_i^d = R^d(p_1^{d-1} \dots p_n^{d-1}) \quad \text{avec} \quad n \geq 2. \quad (4.7)$$

Dans notre approche, les sous-répétitions $\{p_1^{d-1} \dots p_n^{d-1}\}$ doivent être similaires et régulièrement espacées d'une distance δ . La similarité entre deux répétitions est alors définie par comparaison de leurs paramètres, ainsi que de ceux de leur entité de référence. Selon le type de motif associé, d'autres paramètres doivent être comparés comme par exemple l'orientation pour une droite ou le rayon pour un cercle. Notons en particulier qu'une répétition de degré 0 correspond ici à une entité.

Dans la suite de ce manuscrit, nous pourrons également représenter une répétition sous la forme d'un arbre ou comme une inclusion d'ensembles (Fig. 4.30).

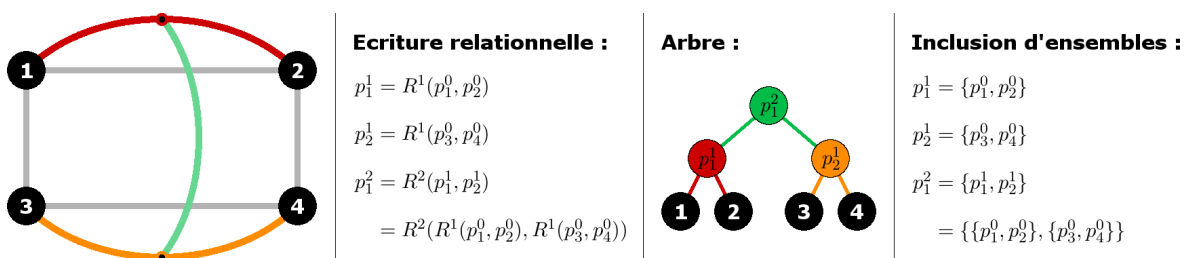


FIGURE 4.30 – Différentes représentations équivalentes d'une répétition.

En reprenant l'exemple du motif rectangulaire illustré dans la figure 4.29, de longueur L et de largeur l , nous pouvons définir deux répétitions équivalentes de second degré :

- en répétant selon L puis l , nous obtenons la répétition $R^2(R^1(p_1^0, p_2^0), R^1(p_3^0, p_4^0))$;
- en répétant selon l puis L , nous obtenons la répétition $R^2(R^1(p_1^0, p_3^0), R^1(p_2^0, p_4^0))$.

Cette équivalence entre plusieurs répétitions candidates peut alors poser un problème d'identification. En effet, même si ces répétitions aboutissent au même positionnement final des primitives dans l'objet, nous ne savons pas *a priori* laquelle a été privilégiée lors de la modélisation.

Nous proposons alors un algorithme de décision ayant pour objectif de ne conserver que les répétitions les plus pertinentes. Celui-ci est basé sur un ensemble de règles prédéfinies, permettant une certaine flexibilité dans son utilisation. Ces dernières peuvent par exemple avoir pour objectif de maximiser la couverture de l'objet par des répétitions, tout en réduisant leur nombre et leur complexité.

Dans notre approche, quatre règles sont définies pour guider la sélection des répétitions les plus pertinentes. Ces dernières sont basées sur des minimisations ou maximisations successives des paramètres de chaque répétition :

1. maximiser le nombre total d'entités regroupées;
2. minimiser le degré;
3. maximiser le nombre d'entités des sous-répétitions;
4. minimiser la distance séparant deux sous-répétitions successives.

Les deux derniers points sont ici évalués en partant d'une feuille de l'arbre, correspondant à l'entité de référence, puis en remontant progressivement jusqu'à la racine (Fig. 4.31). Si les règles de sélection ne permettent pas de départager deux répétitions, par exemple lorsque leurs paramètres sont identiques comme dans le cas d'un motif carré, alors une des deux répétitions peut être choisie arbitrairement.

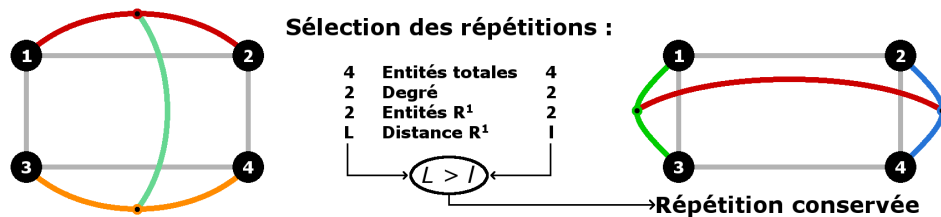


FIGURE 4.31 – Sélection de la répétition la plus pertinente par comparaison de leurs paramètres.

Afin de se rapprocher au mieux d'un processus de modélisation CAO, les répétitions sélectionnées ne doivent pas comporter plus d'une entité en commun. Prenons l'exemple de la figure 4.32.a. Puisque deux répétitions ont été reconnues, cela signifie que l'objet peut être reconstruit en deux temps. La première étape consiste à construire l'entité p_1^0 , puis à la répéter en quatre exemplaires afin de former un carré, représenté ici par la répétition p_2^0 . Ensuite, l'entité p_1^0 est répétée en trois exemplaires le long d'une droite, représentée ici par p_3^0 . Cependant, ces deux répétitions construisent toutes les deux l'entité p_2^0 alors que nous ne souhaitons pas fabriquer deux fois la même entité.

Ainsi, il semble nécessaire de supprimer les répétitions ayant au moins deux entités en commun, en gardant à chaque fois la plus pertinente en fonction des règles prédéfinies. Cela nous permet d'obtenir, par exemple, la configuration de la figure 4.32.b. Cette dernière consiste à répéter p_1^0 pour former un carré, puis à répéter p_2^0 pour créer p_3^0 . Nous obtenons alors un unique exemplaire de chaque entité.

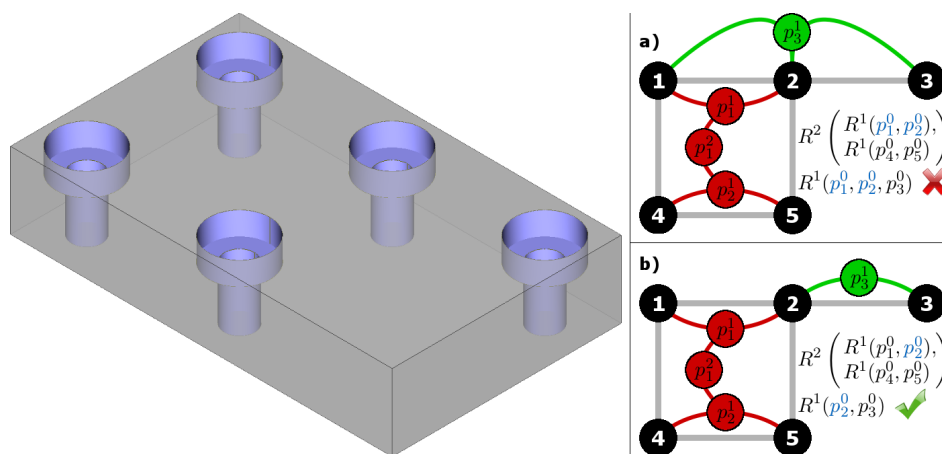


FIGURE 4.32 – Suppression des chevauchements : a) Deux entités en commun, b) Une entité en commun.

D'un autre côté, certaines configurations peuvent être ambiguës ou composées de plusieurs motifs entremêlés. Dans ce cas, plusieurs combinaisons de répétitions peuvent permettre de reconstruire la même pièce. Dans la figure 4.33, nous pouvons voir deux combinaisons possibles

de répétitions permettant de reconstruire la même pièce. Le premier exemple est composé d'une répétition de second degré et d'une répétition de premier degré. Cette combinaison semble maximiser en priorité le nombre maximal d'entités regroupées, et semble donc la plus pertinente selon les quatre règles de sélection décrites précédemment. Le second exemple montre une combinaison de trois répétitions de premier degré. Celle-ci semble minimiser en priorité le degré des répétitions. Ainsi, la seconde combinaison serait sélectionnée en n'utilisant que les trois dernières règles définies précédemment, ce qui montre la flexibilité de l'algorithme.

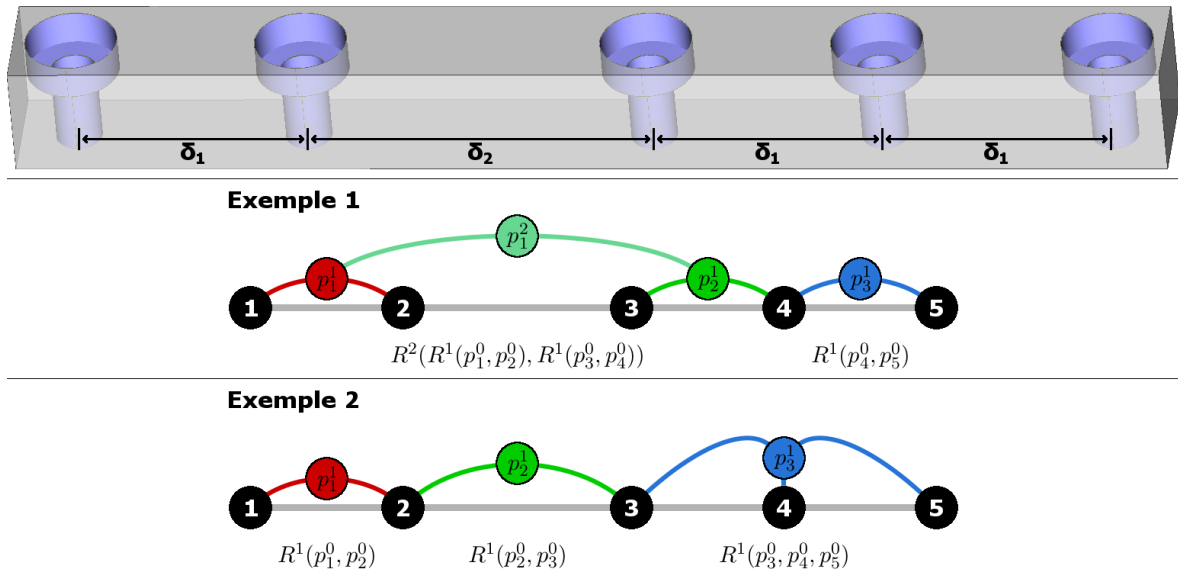


FIGURE 4.33 – Exemple de motif impliquant plusieurs combinaisons possibles de répétitions.

Afin de reconnaître des répétitions en accord avec les règles de sélection prédéfinies, il semble donc nécessaire de prendre en compte toutes les combinaisons possibles de répétitions permettant de reconstruire la pièce. Pour cela, nous proposons un algorithme itératif (Algorithme 7).

La première étape consiste à récupérer l'ensemble des entités $\{p_1^0 \dots p_n^0\}$. Ensuite, une entité p_1^0 est choisie comme référence pour la reconnaissance de répétitions. Nous recherchons alors les répétitions de deux exemplaires, puis celles de trois exemplaires, jusqu'aux répétitions de n exemplaires. Cette recherche est itérée avec chaque entité p_i^0 de l'ensemble de départ, afin d'obtenir toutes les combinaisons.

Ces répétitions forment alors un ensemble de premier degré $\{p_1^1 \dots p_k^1\}$. L'algorithme est ensuite itéré sur ces entités de premier degré, puis de degré supérieur $\{p_1^d \dots p_n^d\}$, jusqu'à ne plus reconnaître de nouvelle répétition. Enfin, une étape de sélection est réalisée afin de retrouver la combinaison de répétitions la plus pertinente. La complexité théorique de cet algorithme peut être très élevée, mais en pratique, les répétitions regroupent généralement peu d'entités.

Algorithme 7 Reconnaissance des combinaisons de répétitions

Soit un ensemble de répétitions $\mathcal{R} \leftarrow \{p_1^0 \dots p_n^0\}$

Soit un degré de répétition $d \leftarrow 0$

répéter

pour tout répétition $p_i^d \in \{p_1^d \dots p_n^d\} \subseteq \mathcal{R}$ **faire**

pour tout nombre d'exemplaires $m \in [2; n]$ **faire**

$\mathcal{R} \leftarrow \mathcal{R} \cup$ Ensemble des répétitions $\{p_1^{d+1} \dots p_k^{d+1}\}$ composées de m exemplaires de p_i^d

fin pour

fin pour

$d \leftarrow d + 1$

jusqu'à ne plus trouver de nouvelle répétition

$\mathcal{R} \leftarrow$ Sélection des répétitions les plus pertinentes de \mathcal{R}

renvoyer \mathcal{R}

La figure 4.34 illustre le déroulement de cet algorithme sur l'objet de la figure 4.33. Lors de l'analyse de premier degré, onze répétitions sont reconnues, dont une seule contient trois entités régulièrement espacées. Lors de l'analyse de second degré, cinq nouvelles répétitions sont reconnues par regroupement des répétitions de premier degré, lorsque ces dernières sont similaires. Aucune répétition de troisième degré n'est reconnue et l'algorithme s'arrête. En effet, les répétitions de second degré ne sont pas paramétrées par des distances identiques, et ne peuvent donc pas être regroupées.

Enfin, la sélection des répétitions les plus pertinentes analyse les seize répétitions obtenues. Notre premier critère étant de maximiser le nombre total d'entités par répétition, l'algorithme commence par sélectionner les répétitions $\{p_2^2, p_3^2, p_4^2, p_5^2\}$, qui regroupent toutes quatre entités. Le second critère ne permet pas de départager ces répétitions, toutes de second degré. Il en va de même pour le troisième, puisque leurs sous-répétitions regroupent toutes deux entités. Le dernier critère consiste à comparer les distances entre les entités pour chaque degré. Au premier degré, ce sont les sous-répétitions p_1^1, p_2^1 et p_3^1 qui minimisent la distance entre deux entités. Il reste donc à départager les répétitions de second degré dont elles font partie, à savoir p_4^2 et p_5^2 . La distance au second degré est alors analysée, et c'est la répétition p_4^2 qui minimise cette distance.

Une seconde passe est ensuite réalisée, et seules les répétitions $\{p_3^1, p_5^1, p_9^1, p_{10}^1\}$ ne partagent pas plus d'une entité avec la répétition p_4^2 . Ici encore, le critère de distance est déterminant et permet de sélectionner la répétition p_3^1 . L'algorithme de sélection fournit alors les répétitions p_4^2 et p_3^1 , ce qui correspond bien à la première combinaison de la figure 4.33.

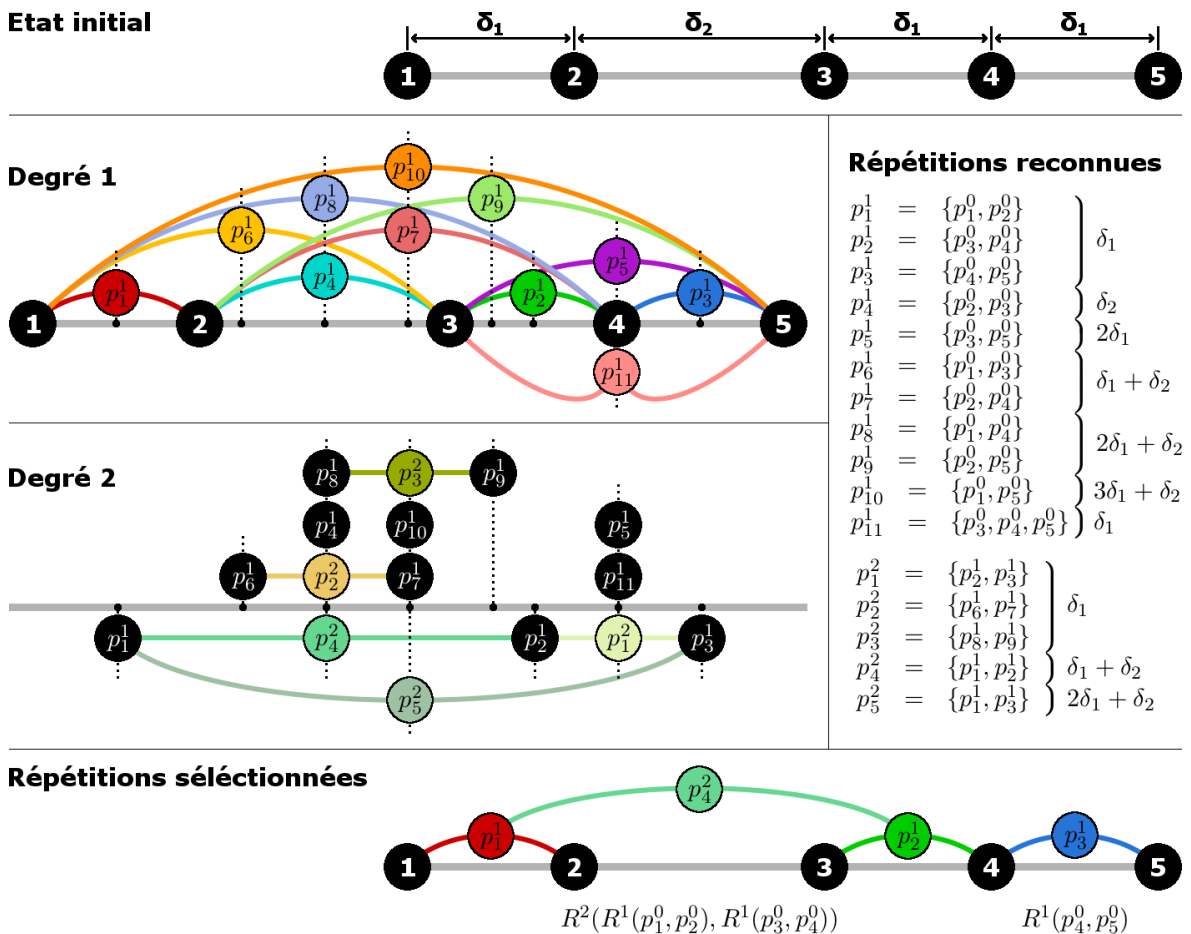


FIGURE 4.34 – Reconnaissance de toutes les combinaisons de répétitions, pour chaque degré.

Nous pouvons remarquer que dans cet exemple, la répétition p_2^2 est équivalente à la répétition p_{11}^1 . En effet, ces deux répétitions sont paramétrées par une unique distance δ_1 . De plus, les sous-répétitions de p_2^2 ont une entité en commun. Cette notion d'équivalence pourrait permettre, par exemple, de réduire la complexité de l'algorithme.

4.4.3 Application à l'embellissement

La reconnaissance de motifs permet de regrouper des primitives, entités ou répétitions similaires. Cependant, dans le cadre de nos travaux, les différents composants de ces motifs ne sont pas identiques à cause des imprécisions contenues dans le maillage, issu d'une numérisation, ayant permis d'extraire les primitives (Fig. 4.35). Il semble alors intéressant d'embellir le modèle en régularisant les paramètres de ces motifs.

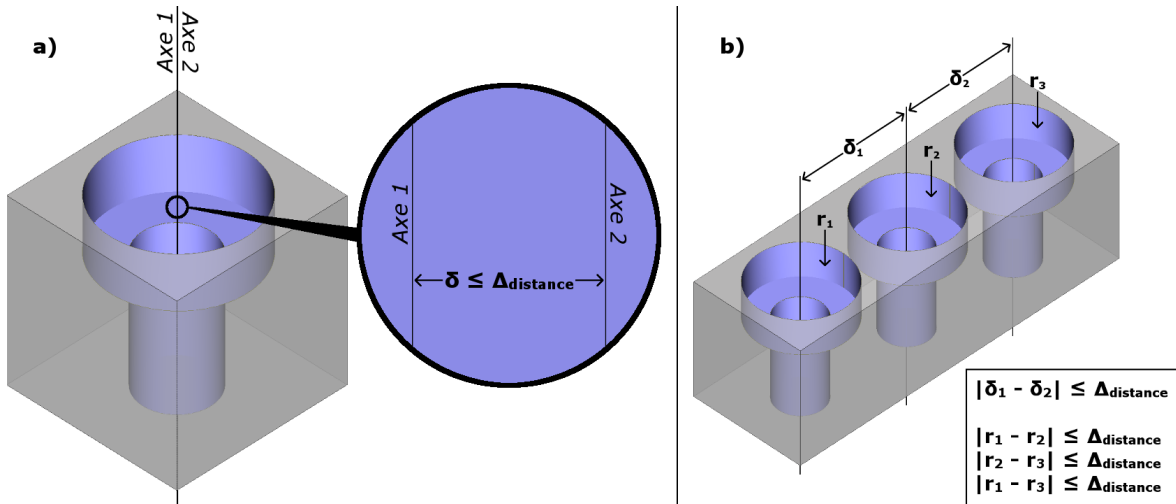


FIGURE 4.35 – Exemple d'une entité et d'une répétition imprécises : a) Trou lamé avec deux axes distincts provenant de ses cylindres, b) Répétition avec des entités séparées par une distance variable.

4.4.3.1 Régularisation d'une entité

Comme expliqué précédemment, une entité est définie par plusieurs paramètres tels qu'une orientation, une position et des dimensions. Cependant, les primitives sous-jacentes sont imprécises dans le cadre de nos travaux. Ces dernières ne respectent alors pas exactement les relations géométriques induites par le processus de modélisation de l'entité. Il est à noter qu'ici, les orientations sont régularisées au préalable afin que les axes soient exactement parallèles (voir section 4.3.1). Par exemple, un trou lamé peut contenir deux axes distincts, provenant des deux cylindres approximativement concentriques (Fig. 4.35.a).

Dans ce cas, il est nécessaire de bien définir les paramètres de l'entité, en particulier lorsqu'ils sont censés être communs à plusieurs primitives. Ainsi, l'axe d'un trou lamé est calculé par une moyenne pondérée des axes provenant de ces deux cylindres. Il en va de même pour le trou fraisé, dont les deux axes proviennent d'un cylindre et d'un cône. Pour ces deux entités, les autres paramètres tels que les dimensions sont directement extraits des primitives correspondantes.

Les paramètres définissant une entité peuvent ensuite être régularisés avec les mêmes algorithmes que pour les primitives seules, présentés dans la section 4.3. Enfin, il est nécessaire de mettre à jour les paramètres des primitives sous-jacentes à chaque entité régularisée. Par exemple, l'axe d'un cylindre appartenant à un trou lamé ou un trou fraisé est remplacé par l'axe de l'entité régularisée.

4.4.3.2 Régularisation d'une répétition

Comme expliqué précédemment, une répétition est définie par plusieurs paramètres tels qu'une entité de référence et une distance séparant deux exemplaires successifs de cette entité. Cependant, l'imprécision des primitives sous-jacentes peut ici aussi complexifier le calcul de ces paramètres (Fig. 4.35.b). En effet, une répétition regroupe plusieurs entités similaires mais non identiques. De même, la distance séparant deux entités successives peut varier lorsque la répétition regroupe au moins trois entités.

Dans ce cas, il est nécessaire de bien définir les paramètres de la répétition. Ainsi, les paramètres de l'entité de référence d'une répétition peuvent être, par exemple, définis par une moyenne pondérée des paramètres de toutes les entités regroupées. De même, la distance séparant deux entités successives peut être calculée en fonction de toutes les distances présentes au sein de la répétition.

Les paramètres définissant une répétition peuvent ensuite être régularisés avec les mêmes algorithmes que pour les primitives seules, présentés dans la section 4.3. Enfin, il est nécessaire de mettre à jour les paramètres des entités sous-jacentes à chaque répétition régularisée. Ainsi, la position de chaque entité est calculée de manière relative au centre de la répétition, et tous ses autres paramètres sont remplacés par ceux de l'entité de référence régularisée au préalable.

4.4.3.3 Analyse des plans de référence

Grâce à la reconnaissance des répétitions, il est possible de distinguer deux types de plans de référence. Les premiers, dits *internes*, permettent d'aligner des primitives provenant toutes d'une seule et même répétition. Les autres, dits *externes*, regroupent des primitives isolées ou provenant de plusieurs répétitions. La figure 4.36 illustre ces deux catégories de plans de référence.

Comme nous l'avons vu dans la section précédente (voir Fig 4.33), deux répétitions sont reconnues à partir de l'objet de la figure 4.36.a. La première regroupe quatre trous lamés formant un carré. Or, un carré implique six plans de référence, correspondant aux quatre côtés et aux deux diagonales (Fig. 4.36.b). La seconde répétition ajoute un cinquième trou lamé, aligné avec un côté du carré. Ce cinquième perçage est alors aligné avec trois des quatre précédents, au travers de deux nouveaux plans de référence. Ces deux plans, alignant des composants de deux répétitions distinctes, sont alors définis comme externes (Fig. 4.36.b).

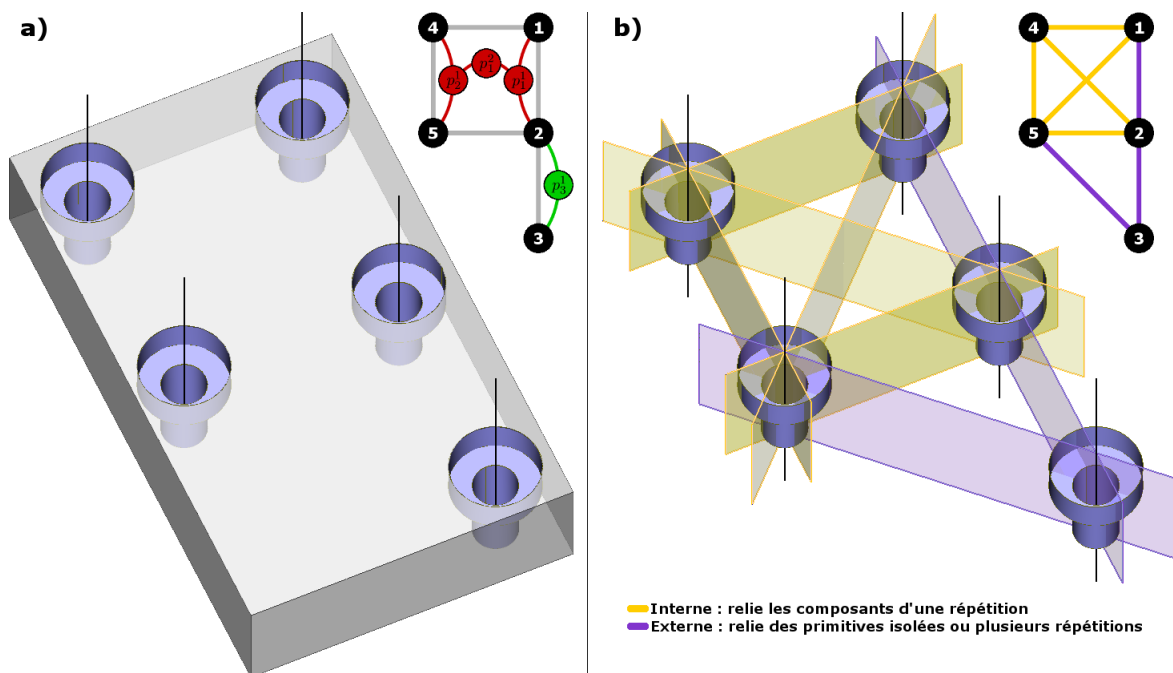


FIGURE 4.36 – Catégorisation des plans de référence : a) Entités et répétitions reconnues (voir Fig. 4.32), b) Plans de référence extraits, définis comme *internes* ou *externes* selon les répétitions reconnues.

Cette analyse permet alors de contraindre la forme des motifs construits grâce aux répétitions, à partir des relations d'angle données par les plans de référence internes. Par exemple, l'inclinaison des diagonales d'un motif rectangle implique un ratio particulier entre la longueur et la largeur du rectangle. D'un autre côté, les plans de référence externes définissent des positions relatives entre les répétitions, permettant ainsi une régularisation plus globale.

4.4.4 Résultats expérimentaux

Dans cette section, nous présentons des résultats de notre reconnaissance d'entités et de répétitions à partir de plusieurs maillages complexes. Sur la figure 4.37.b, nous pouvons voir que quatre entités sont extraites à partir de l'objet *Moldy*. Ces quatre entités sont des lamages parallèles, chacun définis par deux rayons et deux hauteurs. Les quatre lamages ont un rayon de perçage de 0.187 et un rayon de chambrage de 0.287. Les hauteurs correspondantes se situent respectivement dans les intervalles [0.363;0.364] et [0.506;0.508]. Les quatre entités sont similaires, et il est donc possible de les associer pour en déduire des répétitions. Ainsi, le processus d'analyse et de sélection des répétitions fournit une seule répétition pertinente, de second degré (Fig. 4.37.c). Cette dernière consiste à aligner deux répétitions de premier degré, regroupant chacun deux lamages (Fig. 4.37.d). Nous remarquons que les quatre lamages sont disposés en carré, correspondant certainement à la conception originale de la pièce. De plus, le centre de la répétition obtenue est très proche du centre de la pièce, ce qui permet de régulariser la position de la répétition. Enfin, toutes les positions des primitives regroupées sont régularisées en une seule fois, en fonction d'un unique paramètre définissant le côté du carré reconstruit. Dans ce cas, la longueur d'un côté semble être égale à 5.25.

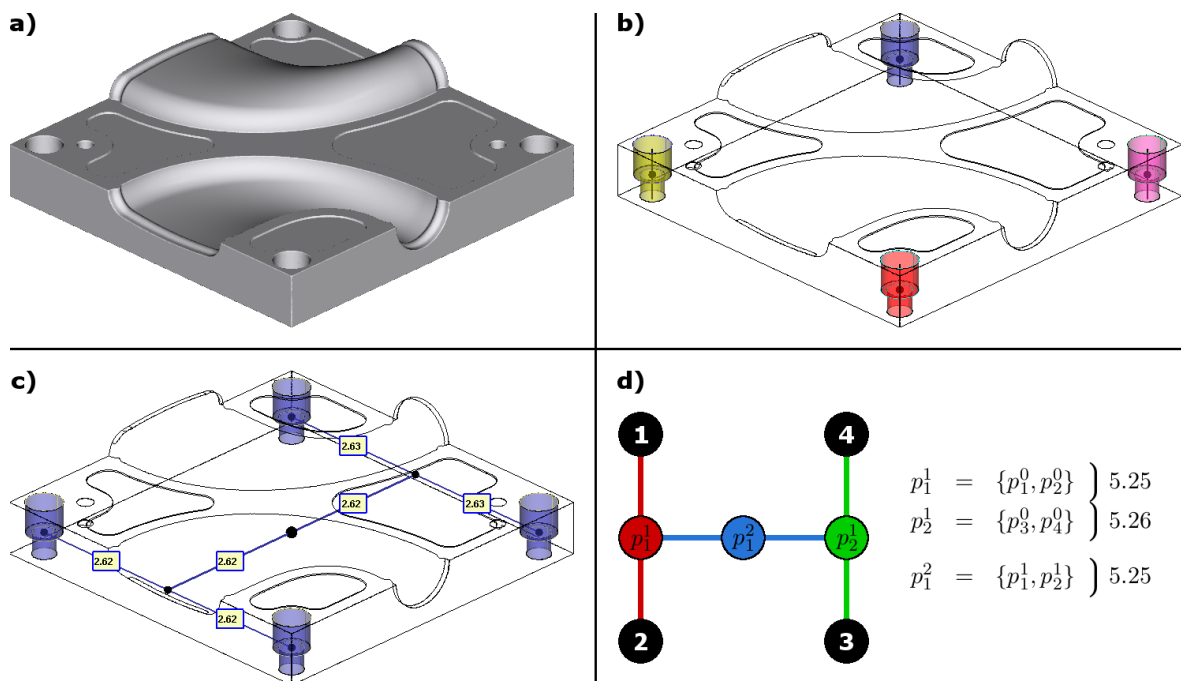


FIGURE 4.37 – Entités et répétitions extraites à partir de *Moldy* : a) Maillage 3D, b) Entités, c) Répétitions, d) Représentation schématique des répétitions.

La figure 4.38 présente nos résultats sur l'objet *Watertight*. Cette pièce contient huit lamages parallèles (Fig. 4.38.b). Les rayons de perçage sont situés dans l'intervalle [4.95;4.97] et les rayons de chambrage dans [6.59;6.96]. Pour les hauteurs, nous trouvons les intervalles [10.23;10.47] et [12.27;13.05]. Les huit lamages sont donc similaires. A partir de ces derniers, deux répétitions de second degré sont extraites. Celles-ci regroupent chacune quatre lamages, selon un motif rectangulaire (Fig. 4.38.c). La figure 4.38.d illustre les répétitions obtenues sous forme schématique. D'après les paramètres de distance obtenus, ces répétitions semblent correspondre à deux rectangles de longueur 82 et de largeur 20. Les positions des entités regroupées peuvent alors être régularisées à partir de ces distances et des centres des répétitions. De plus, nous remarquons que le centre de chaque répétition est approximativement aligné avec le centre de la pièce. Il est alors possible de régulariser leur position, par projection sur l'axe défini par le centre de la pièce et l'orientation du plan contenant la répétition.

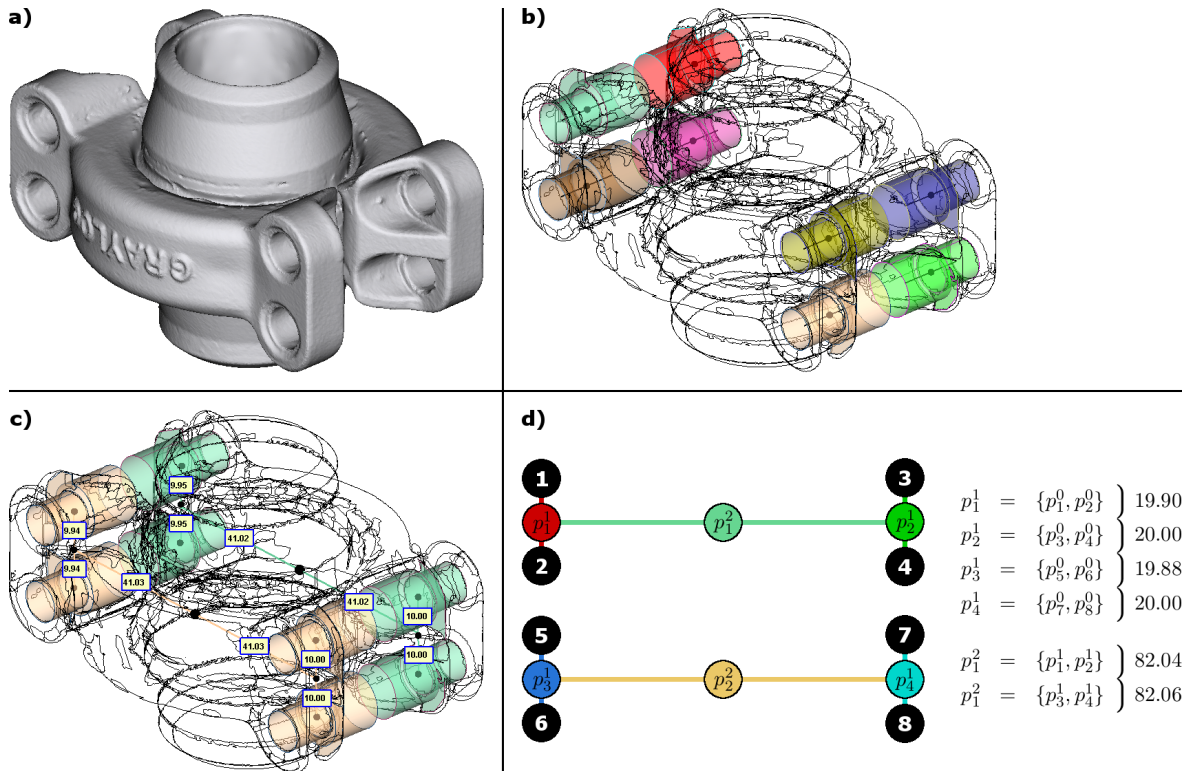


FIGURE 4.38 – Entités et répétitions extraites à partir de *Watertight* : a) Maillage 3D, b) Entités, c) Répétitions, d) Représentation schématique des répétitions.

Cependant, ces deux répétitions ne sont pas regroupées alors qu'elle semblent très similaires. En effet, seule les orientations des entités regroupées diffèrent, et en particulier sont anti-colinéaires. En fait, il s'agit ici d'un processus de modélisation de type *miroir*, qui consiste à répéter une sous-partie de la pièce par une symétrie plane. Nous proposons alors d'adapter les règles de regroupement permettant de construire des répétitions afin de détecter cette configuration. Pour cela, il suffit de définir une nouvelle règle de similarité partielle, consistant à détecter une inversion d'orientation. Ainsi, nous obtenons une répétition de troisième degré, illustrée dans la figure 4.39. Nous remarquons alors que le centre de cette répétition est très proche du centre de la pièce. Dans ce cas, nous pouvons régulariser l'ensemble des primitives regroupées en fonction du centre de la pièce et de trois paramètres de distance.

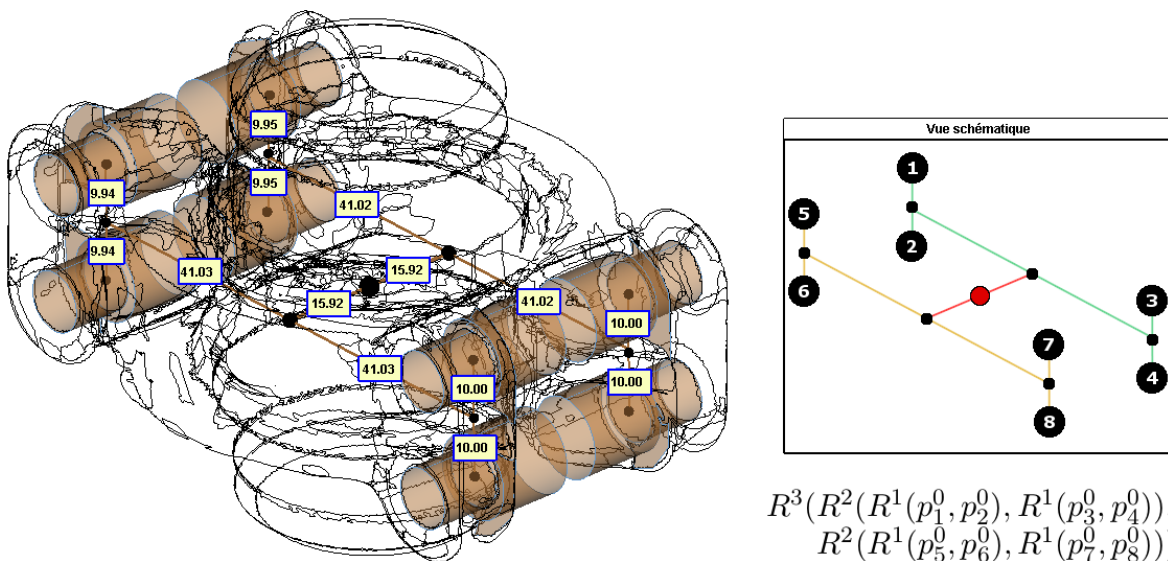


FIGURE 4.39 – Répétition *miroir* extraite à partir des répétitions de *Watertight*, illustrées dans la figure 4.38.

4.5 Expérience d'embellissement intégré à la chaîne de traitement

Dans cette section, nous cherchons à mesurer les déviations entre un modèle original et des primitives extraites à partir d'un maillage. Pour cela, nous proposons de nous placer dans le cadre d'un processus de fabrication et de rétro-ingénierie, comme illustré dans la figure 4.40. Nous avons ainsi conçu un modèle paramétrique à l'aide d'un logiciel de modélisation CAO. Ensuite, nous avons fabriqué la pièce à l'aide d'une imprimante 3D, puis l'avons numérisée grâce à un scanner à lumière structurée. Enfin, nous avons appliqué notre processus de rétro-ingénierie au maillage obtenu. Le modèle original et le modèle reconstruit peuvent alors être comparés.

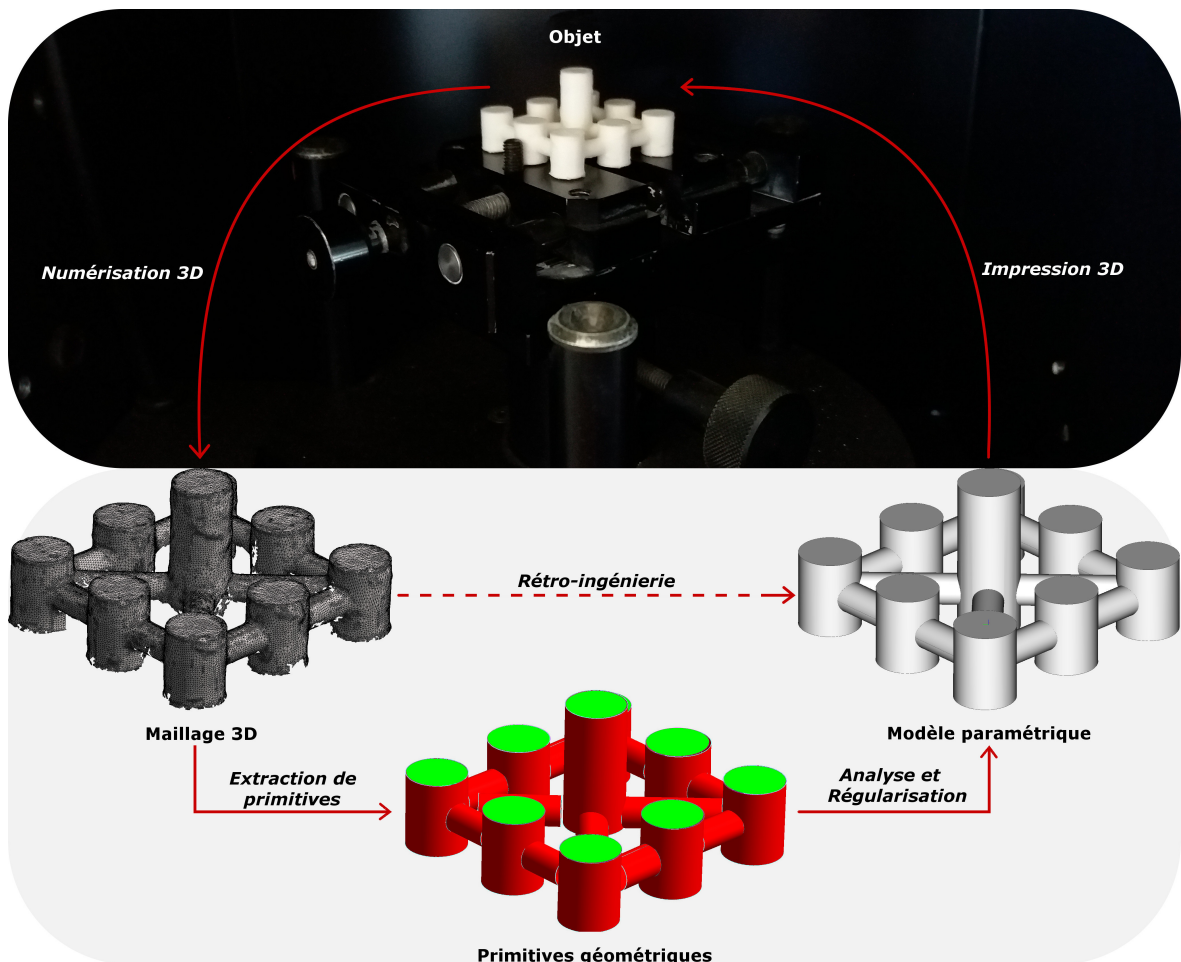


FIGURE 4.40 – Processus de fabrication et de rétro-ingénierie, à base d'impressions et de numérisations 3D.

La figure 4.41 présente l'objet *Lattice* sous différents angles de vue. Ce modèle contient neuf cylindres parallèles et de même rayon, disposés régulièrement sur une grille carrée. Ces neuf cylindres sont ensuite reliés par d'autres cylindres, orthogonaux aux premiers, formant des angles de 90° ou 45° par rapport au repère orthonormé. La figure 4.41.a montre en particulier que le modèle original a été conçu en suivant les axes du repère orthonormé (O_x, O_y, O_z) . D'un autre côté, la figure 4.41.b montre que le maillage obtenu par numérisation est positionné différemment. Afin de comparer correctement les deux modèles, il est donc nécessaire de définir un repère orthonormé correspondant au maillage. Pour cela, nous nous basons sur le repère de référence extrait à partir des primitives lors de l'analyse des orientations (voir section 4.3.1). La figure 4.41.c montre que le repère estimé permet de retrouver le positionnement du modèle original.

Notre objectif est de retrouver des primitives, à partir du maillage issu de numérisation, puis de les régulariser pour respecter les relations géométriques présentes dans le modèle original. Ces dernières sont essentiellement des relations d'orientation (parallélisme, orthogonalité et inclinaison à 45°), de dimensions (égalité entre les rayons et hauteurs des cylindres) et de positions (disposition régulière sur une grille). Mais le maillage ne contient pas la partie inférieure du modèle,

non visible lors de la numérisation. Dans ce cas, les neuf plans correspondant ne sont pas pris en compte lors de la comparaison des modèles.

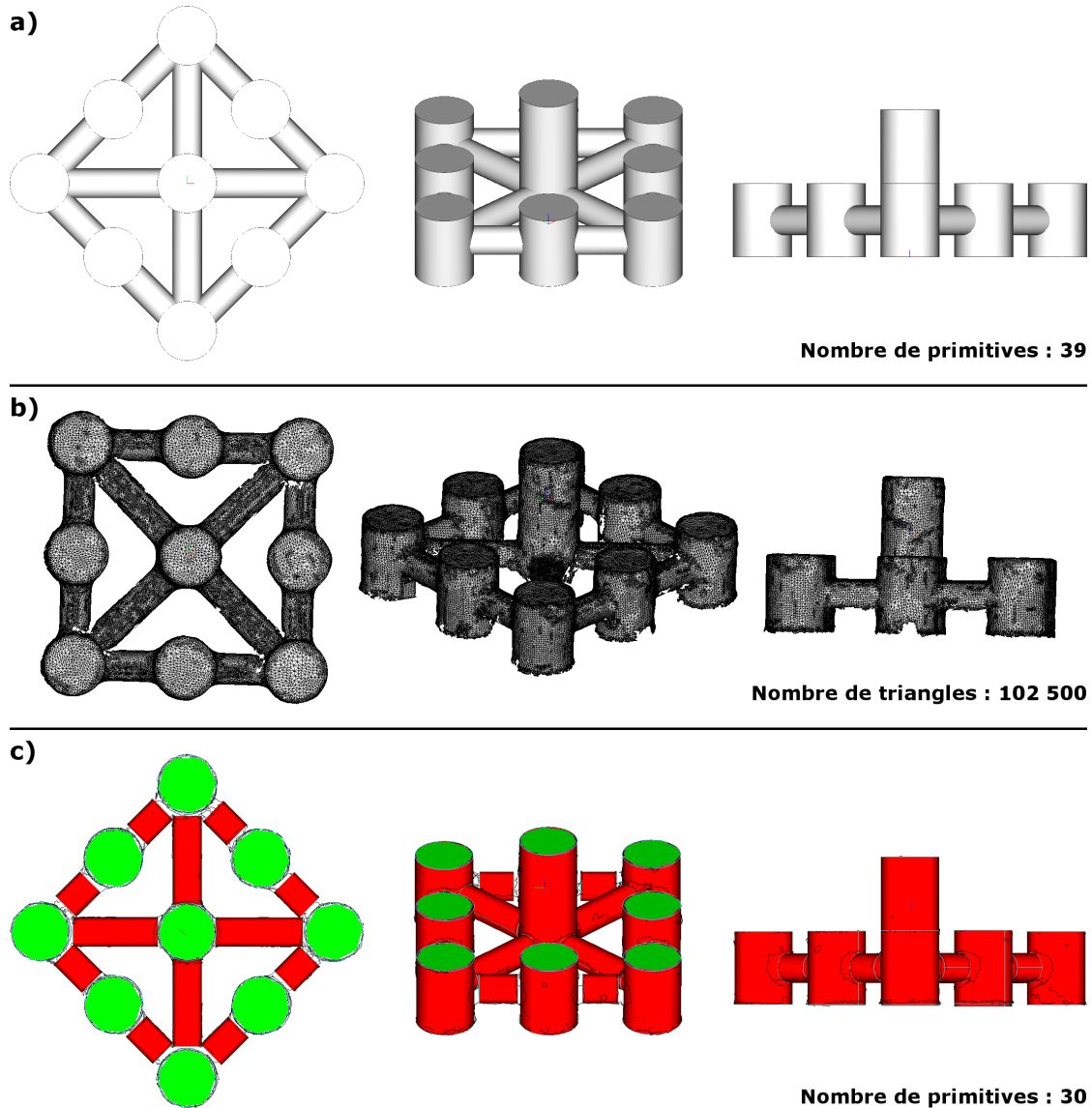


FIGURE 4.41 – *Lattice* sous différents points de vue : a) Modèle original dans le repère CAO, b) Maillage issu de numérisation dans le repère du scanner 3D, c) Primitives dans le repère de référence estimé.

Ainsi, nous appliquons notre processus de régularisation sur l'ensemble des primitives extraites (Fig. 4.42). Lors de l'analyse, cinq groupes de primitives parallèles sont détectés (Fig. 4.42.a). Le premier contient neuf cylindres et neuf plans, et correspond à l'axe principal de la pièce. Les second et troisième groupes contiennent les quatre cylindres formant la croix centrale de *Lattice* et correspondent aux deux autres axes du repère de référence. Les deux derniers groupes rassemblent les huit cylindres restant, dont les axes sont définis à 45° du repère de référence. Il est aussi à noter que l'orientation du premier groupe est orthogonale à celles de tous les autres.

Ensuite, deux groupes de rayons similaires sont détectés (Fig. 4.42.b). Le premier regroupe les neuf cylindres les plus larges, dont les rayons se situent dans l'intervalle $[3.94;3.98]$. Le second contient les douze plus étroits, avec des rayons dans l'intervalle $[1.97;2.00]$. En définissant une règle métier, par exemple en imposant des rayons multiples de 0.25, nous obtenons deux groupes de cylindres régularisés de rayon 4.00 et 2.00. De même, plusieurs groupes de hauteurs sont régularisés : huit cylindres de hauteur 10, huit autres de hauteur 5, quatre de hauteur 12 et enfin le cylindre central de hauteur 15. Les dimensions obtenues correspondent alors exactement à celles du modèle original.

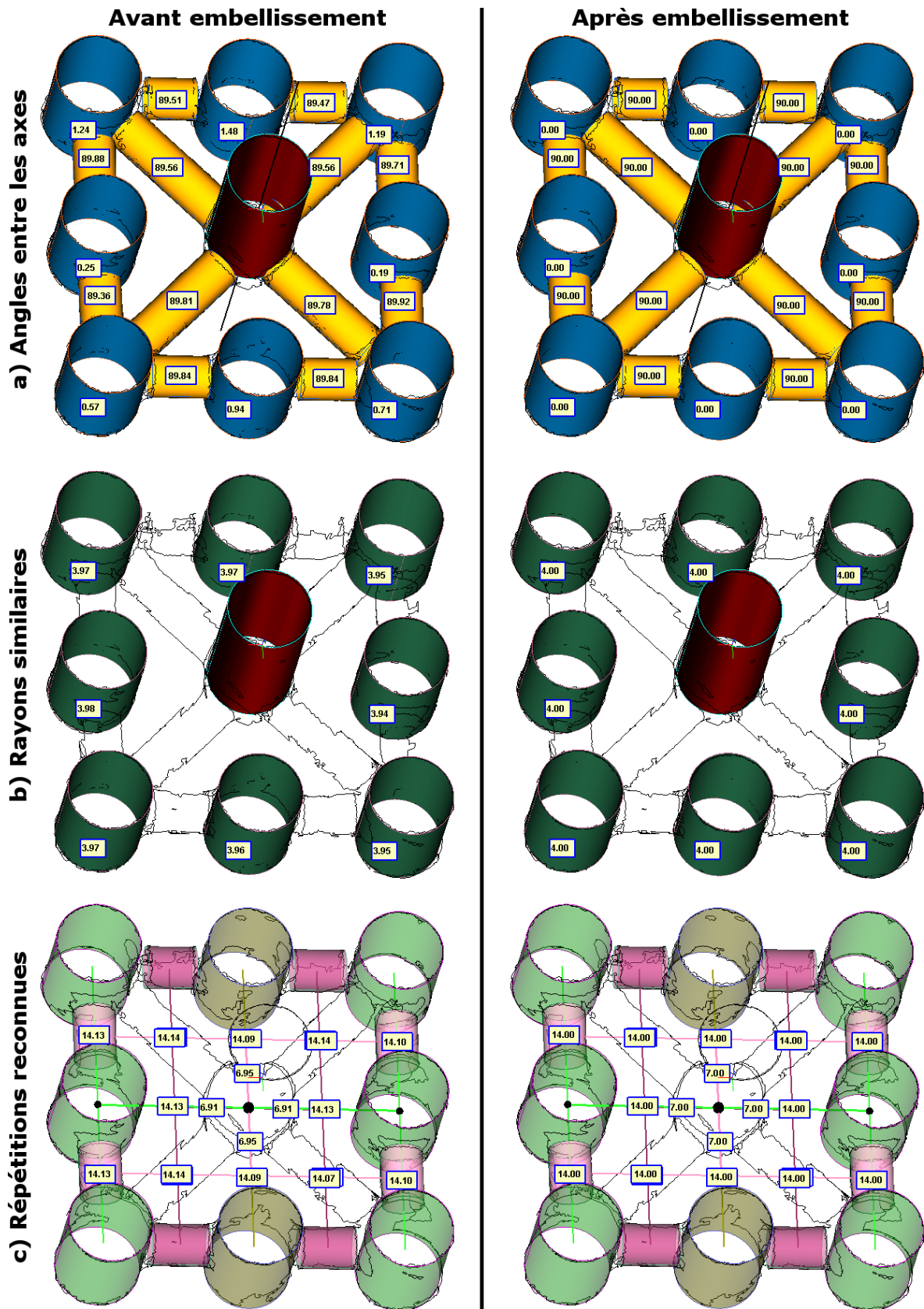


FIGURE 4.42 – Régularisation des primitives de *Lattice* :

- a) Orientations : les cylindres régularisés sont exactement parallèles, orthogonaux et inclinés à 45° ;
- b) Dimensions : les cylindres régularisés sont paramétrés par des rayons identiques ;
- c) Positions : les répétitions reconnues permettent de contraindre toutes les positions des primitives.

Enfin, les positions des primitives sont analysées à travers la reconnaissance de plans de référence et de répétitions (Fig. 4.42.c). Ainsi, cinq répétitions sont reconnues, dont une de premier degré et quatre de second degré. Les distances paramétrant ces répétitions se situent dans les intervalles [13.82; 13.9] et [28.14; 28.34]. En définissant une nouvelle règle métier, par exemple des écartements entiers, ces distances deviennent respectivement 14 et 28. En particulier, les centres des répétitions coïncident avec le centre de la pièce, permettant une régularisation plus fine. La figure 4.43 compare ainsi les plans de construction du modèle, avant et après régularisation. Dans ce cas, le plan CAO après régularisation correspond exactement à celui du modèle original.

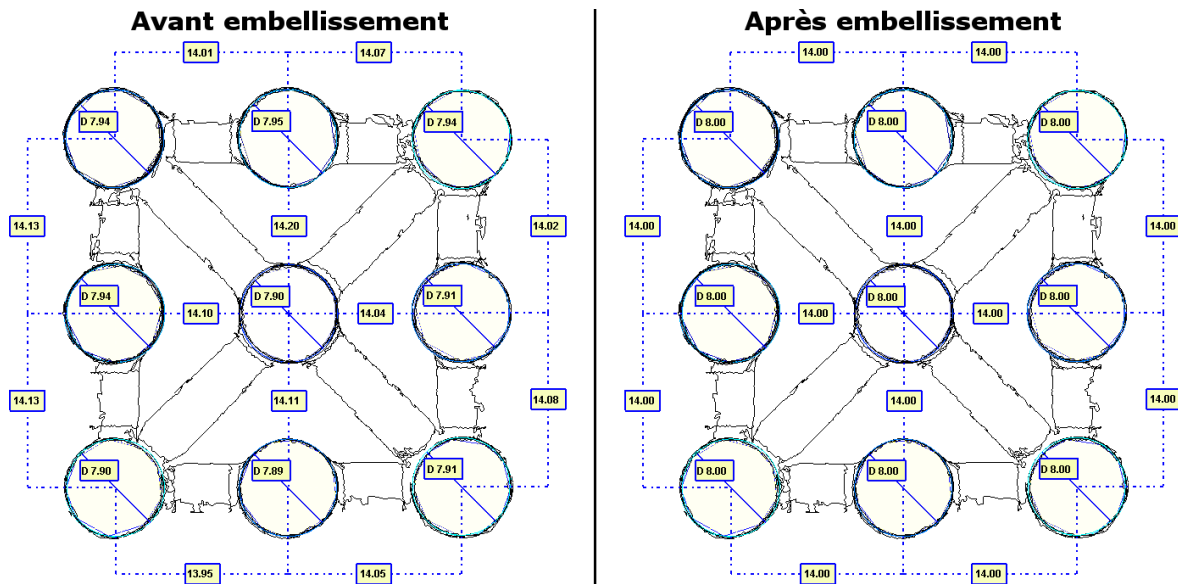


FIGURE 4.43 – Plan CAO du modèle reconstruit de *Lattice*, avant et après analyse et régularisation.

Cependant, comme décrit précédemment, nous avons dû définir des règles métier particulières pour aboutir à ce résultat, notamment pour les aspects dimensionnels de la pièce. En effet, nous ne connaissons *a priori* ni l'échelle de la pièce numérisée, ni l'unité de mesure (millimètres ou pouces par exemple). De même, il semble compliqué de prendre en compte toutes les mesures possibles, pouvant même parfois correspondre à des nombres rationnels.

La figure 4.44 présente une seconde pièce modélisée, *TestPlate*, imprimée puis numérisée. Celle-ci comporte plusieurs extrusions positives et négatives, de différentes dimensions. En particulier, les primitives recherchées sont placées régulièrement le long des axes du repère ortho-normé. Il est à noter que ces primitives sont volontairement positionnées selon des coordonnées non-entières et aux décimales infinies, correspondant à une fraction de longueur de la pièce. Nous obtenons alors des positions relatives par rapport au centre de la pièce, par exemple $\frac{1}{2}$ ou $\frac{2}{3}$.

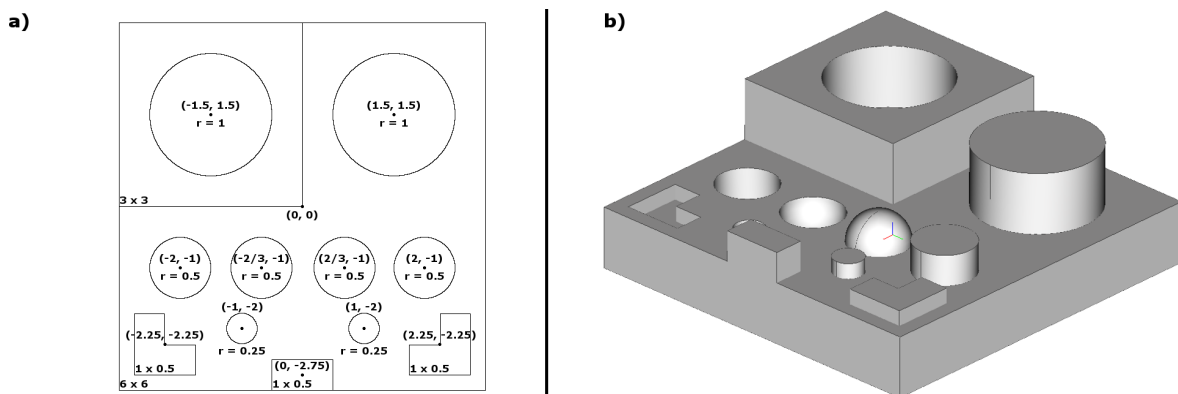


FIGURE 4.44 – Modélisation de *TestPlate* : a) Plan 2D original, b) Modélisation 3D du modèle.

Pour obtenir une qualité suffisante de numérisation, la pièce a été imprimée à un facteur 10. La figure 4.45 présente le maillage obtenu après numérisation, ainsi que les primitives extraites. Nous pouvons remarquer que le maillage est très imprécis et contient même des artefacts de numérisation. L'extraction de primitives fournit tout de même des résultats satisfaisants, bien que les paramètres de ces dernières soient imprécis.

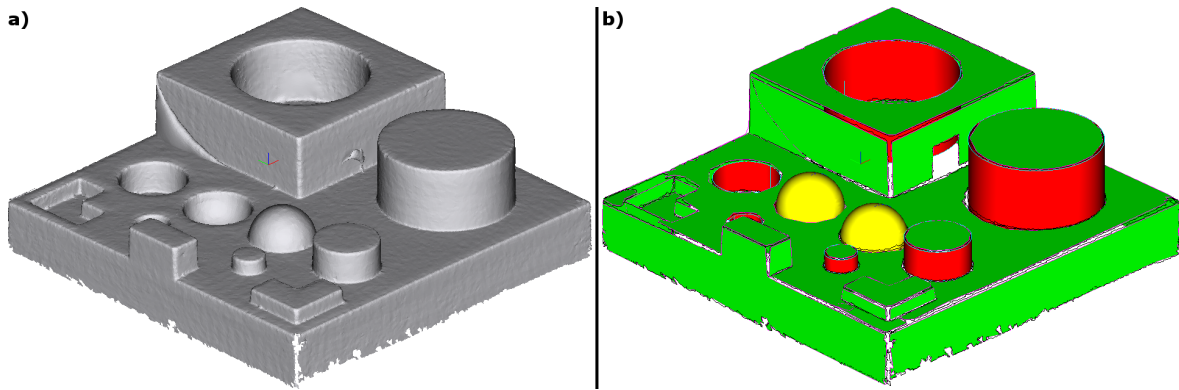


FIGURE 4.45 – Rétro-ingénierie de *TestPlate* : a) Maillage numérisé après impression, b) Primitives extraites.

Nous avons appliqué notre processus d'analyse et de régularisation à l'ensemble de primitives obtenus. Dans la figure 4.46, le plan CAO du modèle régularisé semble alors correspondre exactement au plan du modèle original multiplié par un facteur 10. Cependant, la distance séparant les deux sphères reste imprécise. En effet, cette dernière correspond à une fraction non-entière de la longueur de la pièce. Dans notre cas, cette longueur a été arrondie et ne correspond pas exactement à l'intention de modélisation. Pour prendre en compte cette notion de distances relatives, il est nécessaire d'ajouter de nouvelles règles à notre algorithme.

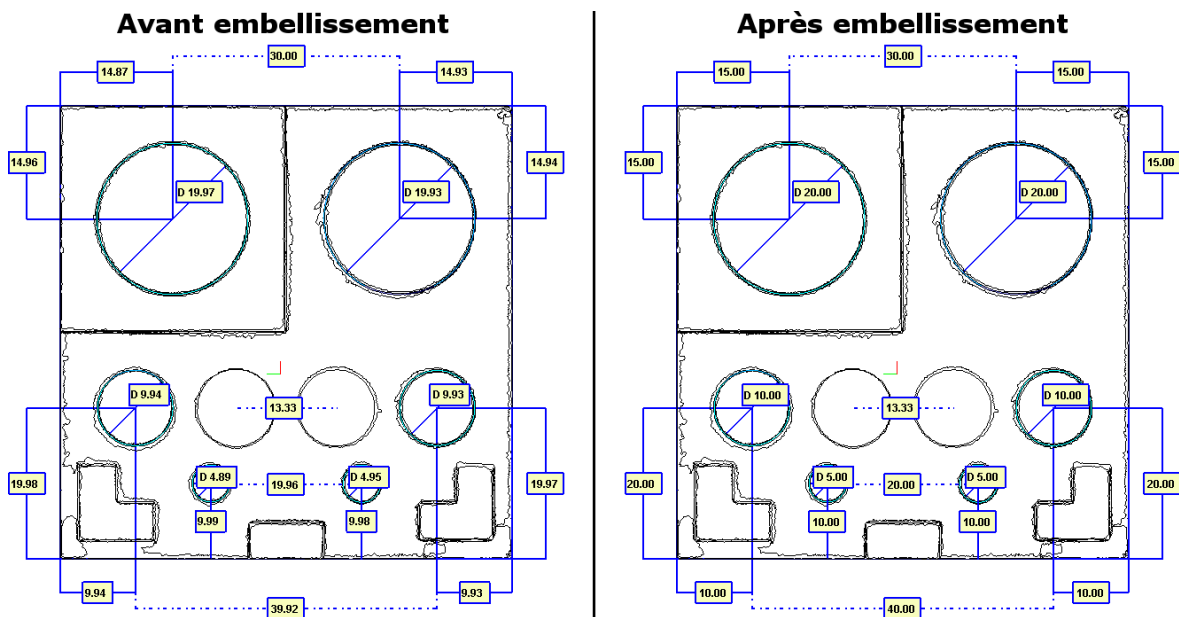


FIGURE 4.46 – Plan CAO du modèle reconstruit de *TestPlate*, avant et après analyse et régularisation.

4.6 Conclusion

Dans ce chapitre, une nouvelle méthode d'analyse hiérarchique a été proposée dans le but de retrouver des relations géométriques entre des primitives extraites à partir d'un maillage 3D issu de numérisation. Mais les processus de fabrication et de numérisation sont imparfaits. Les paramètres des primitives extraites sont alors souvent imprécis. Dans ce cas, il est nécessaire de concevoir des algorithmes robustes afin de reconstruire un modèle final précis.

Dans un premier temps, des relations sont estimées entre les primitives par paires, en fonction de tolérances correspondant à des règles métier. Ensuite, des informations globales sont extraites à partir de l'ensemble des primitives et de leurs relations, ce qui permet de construire un système de référence comme en modélisation CAO. Ce système, constitué d'un repère orthonormé tridimensionnel et de plans de référence, permet alors d'estimer les relations obtenues de manière plus globale, ce qui réduit l'impact des imprécisions. Enfin, ces éléments de référence sont utilisés dans le cadre d'un processus d'embellissement des primitives. Ce dernier consiste à régulariser les paramètres des primitives de sorte qu'elles respectent des relations géométriques, dans le but de reconstruire un modèle final plus fidèle à l'original.

L'estimation d'un système de référence de manière globale est essentielle. En effet, ce dernier permet de définir des relations géométriques globales et fiables, mettant en jeu de nombreuses primitives. Grâce à cela, il est possible d'analyser ou de régulariser les paramètres des primitives indépendamment les unes des autres, en se basant sur des références globales et absolues. En particulier, cette approche globale permet d'éviter certains problèmes tels que la propagation d'erreur. A la fin du processus, l'ensemble des primitives régularisées représente un modèle contraint respectant des relations d'orientations, de positions et de dimensions.

Nous proposons également d'extraire des configurations particulières de primitives, correspondant à des entités ou des répétitions. Cette seconde analyse permet de renforcer des relations ou encore de définir des positions relatives entre les primitives d'un même motif. Il est alors possible de régulariser à nouveau les primitives, d'une manière plus globale et correspondant à un processus de fabrication précis.

La méthode proposée est un premier pas vers un processus de rétro-ingénierie complètement automatique, cherchant à retrouver l'intention de conception originale d'une pièce manufacturée. Cependant, il peut être difficile de prendre en compte toutes les possibilités de placement des primitives au sein d'un objet, c'est-à-dire toutes les distances et proportions possiblement utilisées lors de la modélisation. L'algorithme présenté est alors conçu pour être adaptable et facilement extensible, en ajoutant d'autres règles métier concernant les relations géométriques, les entités et les répétitions.

Les travaux présentés dans ce chapitre ont été valorisés par les publications suivantes :

- Gauthier, S., Puech, W., Bénière, R., Subsol, G. (2018). Orientation Beautification of Reverse Engineered Model. In *GRAPP : International Conference on Computer Graphics Theory and Applications*, 2018(13) : 91-100.
- Gauthier, S., Subsol, G., Bénière, R., Puech, W. (2018). CAD-Driven Analysis of Reverse Engineered Geometric Models : Application for Beautification. *IEEE Transaction on Visualization and Computer Graphics*, soumis en février 2018.



Conclusion et perspectives

Dans ce chapitre, nous présentons un bilan de nos travaux de recherche. Tout d'abord, une procédure complète de rétro-ingénierie résume les différentes étapes décrites dans ce manuscrit. Ensuite, nous présentons des perspectives d'amélioration de ce processus, à différentes étapes. Ainsi, nous souhaiterions détecter de nouvelles primitives comme des surfaces à forme libre pour compléter les modèles reconstruits, ou encore approfondir notre analyse sémantique pour déduire de nouvelles informations sur l'objet.

Résumé de nos contributions

Dans ce manuscrit, nous avons amélioré la procédure de rétro-ingénierie développée dans la thèse de Roseline Bénére [5] dans le but de traiter des maillages complexes issus de numérisation 3D. Nos travaux se focalisent alors sur les deux grandes étapes du processus, à savoir l'extraction de primitives et la reconstruction du modèle. Nous avons ainsi proposé :

- **Une méthode robuste d'extraction de primitives à partir de maillages numérisés (chapitre 3).** Celle-ci consiste à segmenter le maillage de manière robuste pour obtenir une seule primitive par sous-maillage. Il est alors possible d'en extraire un ensemble de primitives dont les paramètres sont robustes aux imprécisions, grâce à des régressions et des optimisations. Dans le cas où certains sous-maillages contiendraient plusieurs primitives, nous avons développé une procédure de calcul automatique des tolérances permettant d'extraire des primitives localement. Notre méthode permet alors de maximiser la couverture d'un maillage 3D par des primitives géométriques.
- **Une nouvelle méthode d'analyse sémantique d'un ensemble de primitives (chapitre 4).** Les primitives extraites à partir d'un maillage numérisé sont généralement imprécises. Elles ne respectent donc pas les relations géométriques comme le parallélisme ou l'orthogonalité, pourtant présentes dans le modèle original. Notre méthode d'analyse extrait alors des informations de conception de la pièce, telles qu'un système de référence ou des motifs particuliers, qui permettent à la fois de caractériser la pièce mais aussi de corriger les paramètres des primitives afin de reconstruire un modèle correspondant au mieux à l'original.

Ces contributions permettent de définir un processus complet de rétro-ingénierie capable de traiter efficacement des maillages issus de numérisation d'objets manufacturés, même complexes. Les résultats obtenus sur de nombreux maillages numérisés ayant des caractéristiques très différentes sont très satisfaisants. En effet, les modèles reconstruits à partir des primitives extraites semblent cohérents par rapport aux modèles paramétriques originaux. De plus, notre méthode est totalement automatique et ne nécessite que la validation du modèle final par l'utilisateur. Notons que la méthode présentée a été implémentée dans l'environnement logiciel de C4W. Ainsi, l'extraction automatique de primitives est incluse et déjà utilisée dans des logiciels distribués par C4W à ses clients. L'analyse sémantique des primitives est en cours d'intégration.

Perspectives

Une des limites de notre méthode de rétro-ingénierie est de ne pas pouvoir reconstruire les zones du maillage non représentables par des primitives. Nos travaux futurs consisteront donc dans un premier temps à compléter notre processus par une étape d'extraction de surfaces à forme libre, comme illustré dans la figure 5.2. Celles-ci pourront alors s'appuyer à la fois sur des contours de primitives ajustées, des relations géométriques et des sous-maillages ne contenant aucune primitive. Dans la littérature, de nombreux travaux proposent par exemple d'approximer une surface à forme libre à partir d'un nuage de points [37, 31] ou d'un maillage [44, 123], et éventuellement selon des contraintes géométriques [35, 72]. Il semble donc intéressant d'utiliser ces méthodes pour maximiser la couverture des maillages par des surfaces continues, et ainsi compléter les modèles reconstruits.

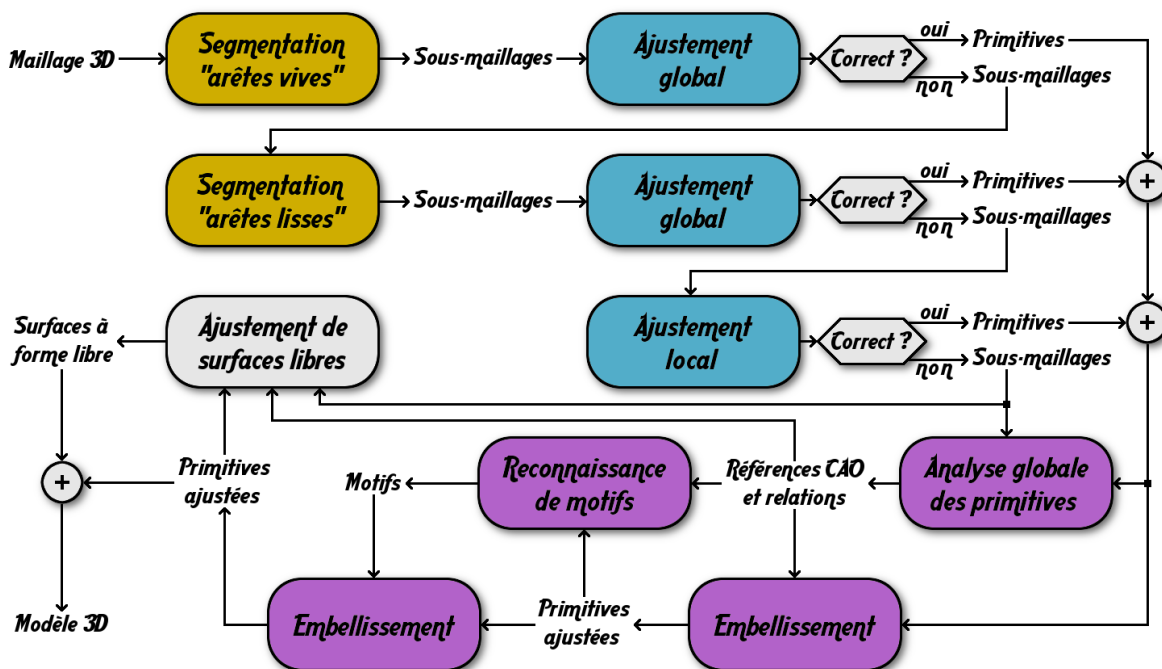


FIGURE 5.2 – Schéma de notre approche pour la rétro-ingénierie avec une étape supplémentaire d'extraction de surfaces à forme libre, dans le but de compléter les modèles reconstruits.

Nous aimerions par la suite optimiser l'implémentation de notre méthode afin de réduire les temps de calcul. En effet, plusieurs étapes de notre méthode pourraient être parallélisées. Par exemple, chaque itération de la segmentation, par arêtes vives ou lisses, est indépendante en calcul et en mémoire. De même pour l'ajustement, global ou local, qui est fait indépendamment pour chaque type de primitive. En partant de cela, nous pourrions grandement accélérer le processus et traiter des maillages de plusieurs millions de triangles en seulement quelques secondes.

Il serait également intéressant de pousser notre analyse sémantique de l'ensemble des primitives afin, par exemple, d'extraire de nouvelles primitives ou de supprimer des primitives existantes qui sont incorrectes du point de vue conception de l'objet. En effet, l'identification de motifs particuliers décrite dans le chapitre 4 peut par exemple permettre de supposer un placement idéal pour une primitive et ainsi guider un processus d'ajustement supplémentaire.

Enfin, comme pointé dans le manuscrit de thèse de Roseline Bénére [5], la méthode de reconstruction des modèles B-Rep implémentée dans la suite logicielle de C4W n'est efficace que lorsque l'ensemble des primitives sont extraites. Dans le cas de numérisation 3D, il est cependant courant d'être confronté à des occultations qui ne permettent pas de reconstruire la totalité d'un objet. Dans ce cas, il serait intéressant de proposer une méthode hybride, prenant en compte à la fois des intersections continues entre primitives lorsque cela est possible et des contours discrets lorsqu'il manque des informations.

Ajustement de surfaces à forme libre

Nous avons réalisé quelques expérimentations afin de montrer l'intérêt de l'extraction de surfaces à forme libre. La figure 5.3 montre un exemple d'application sur le modèle reconstruit à partir du maillage *Moldy*. Après notre processus, il reste deux sous-maillages non représentés par des primitives (Fig. 5.3.a). Les contours de ces sous-maillages sont alors extraits puis découpés en fonction des intersections entre les primitives adjacentes (Fig. 5.3.b). Dans cet exemple, nous obtenons alors quatre contours (représentés de différentes couleurs dans la figure 5.3.b), suggérant une surface à forme libre définie par quatre courbes. Dans ce cas, à l'aide des outils de la suite logicielle de C4W, chaque contour est approximé par une courbe de Bézier ou une B-Spline, puis une surface à forme libre est approximée à partir des quatre courbes obtenues (Fig. 5.3.c). Cette nouvelle surface est alors ajoutée à l'ensemble des primitives. Ainsi, dans la figure 5.3.d, nous pouvons voir que le maillage *Moldy* est entièrement couvert par des surfaces géométriques, ce qui permet au final de reconstruire un modèle complet.

Il serait intéressant par la suite de prendre en compte les points du sous-maillage et pas uniquement ses contours, afin de rectifier la surface approximée. De même, les frontières de la surface devraient être projetées sur les primitives adjacentes correspondantes afin de pouvoir reconstruire un modèle B-Rep fermé. Enfin, des contraintes de continuités pourraient aussi être appliquées sur les surfaces à forme libre à partir des paramètres des primitives adjacentes.

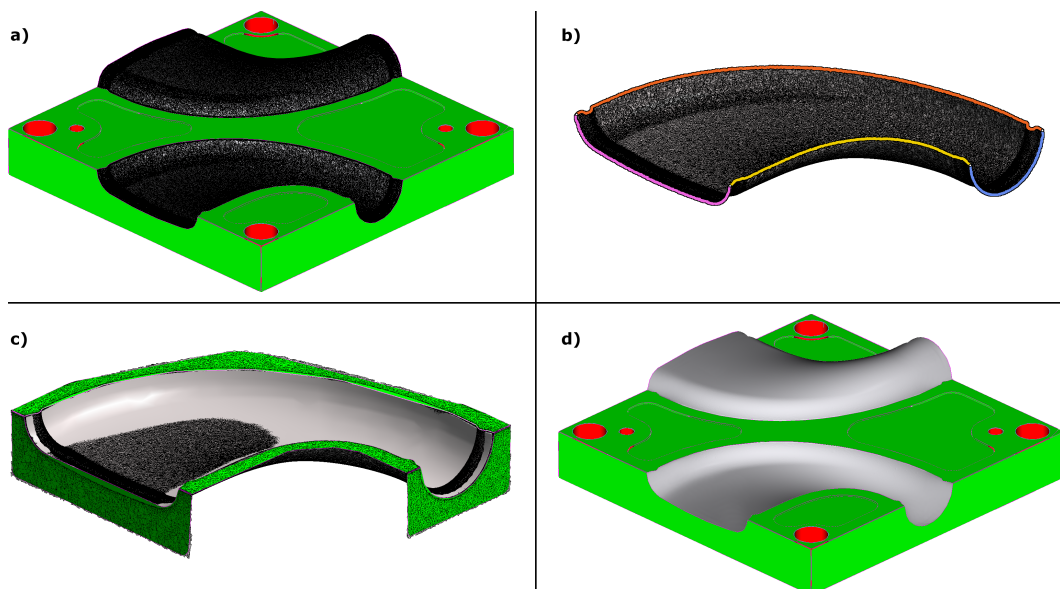


FIGURE 5.3 – Extraction des surfaces à forme libre de *Moldy* : a) Primitives et sous-maillages de type inconnu, b) Sous-maillage de type inconnu et son contour découpé en fonction des primitives adjacentes, c) Surface à forme libre approximée à partir des quatre contours, d) Modèle reconstruit.

La figure 5.4 montre un second cas d'application sur un sous-maillage de *Pump*. Ce dernier représente un tube, non représentable par des primitives simples. Les contours du sous-maillage sont analysés comme précédemment. Dans la figure 5.4.a, nous trouvons deux contours adjacents à des primitives (en cyan), et de nombreux contours correspondant à des occultations lors de la numérisation (en rouge). Seul les deux contours représentant les extrémités du tube sont alors considérés. Ensuite, comme illustré dans la figure 5.4.b, les centres de courbure associés aux points du sous-maillage sont extraits (en vert) puis regroupés selon une distance prédéfinie afin d'obtenir un ensemble de points plus réduit et précis (en rouge). Ces nouveaux points permettent alors d'approximer une courbe de Bézier ou une B-Spline, représentant la directrice du tube (Fig. 5.4.c). Enfin, les valeurs absolues maximales de courbure associées au point du maillage correspondent localement au rayon du tube. Un rayon plutôt constant, comme c'est le cas ici, suggère la présence d'un cylindre généralisé circulaire uniforme (CGCU). Celui-ci est alors reconstruit, à l'aide des outils de modélisation fournis par C4W, à partir de la courbe et du rayon moyen (Fig. 5.4.d).

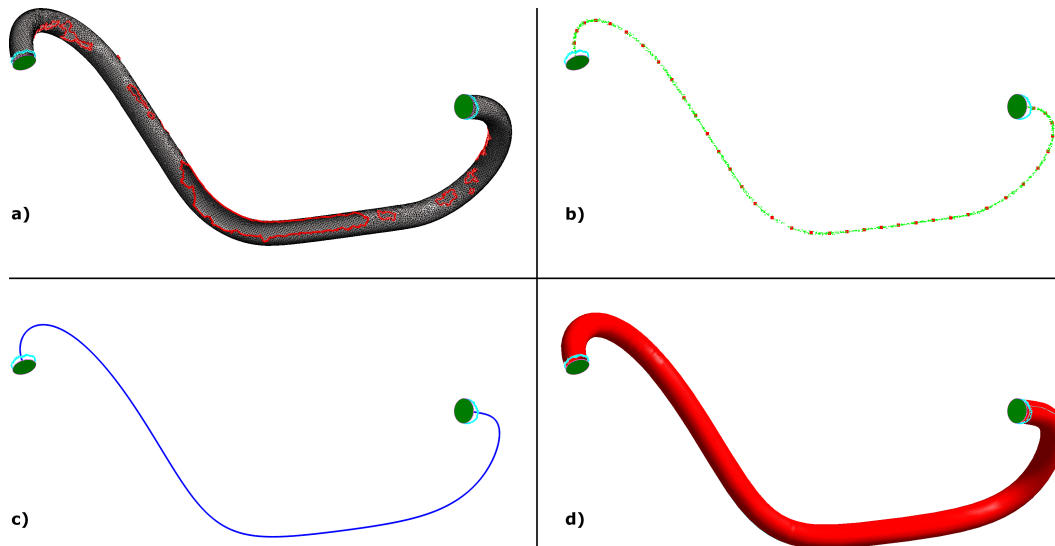


FIGURE 5.4 – Extraction d'un cylindre généralisé circulaire uniforme (CGCU) : a) Primitives et sous-maillage, b) Centres de courbure (en vert) regroupés selon une distance minimale (en rouge), c) Courbe approximée représentant la directrice du CGCU, d) Reconstruction du CGCU avec le rayon de courbure moyen.

Dans le cas où le rayon de courbure n'est pas constant, il serait intéressant d'estimer la fonction qui permet de définir le rayon en chaque point de la courbe directrice. Pour plus de simplicité, nous avons ici opté pour une construction par sections. Ainsi, nous calculons des plans de coupe répartis uniformément le long de la courbe directrice. Ces plans permettent alors de construire des sections 2D du maillage, puis de reconstruire une surface à forme libre à l'aide des outils de modélisation de C4W. La figure 5.5 montre un exemple d'application sur *Vase*, pour lequel la surface obtenue couvre une grande partie du maillage.

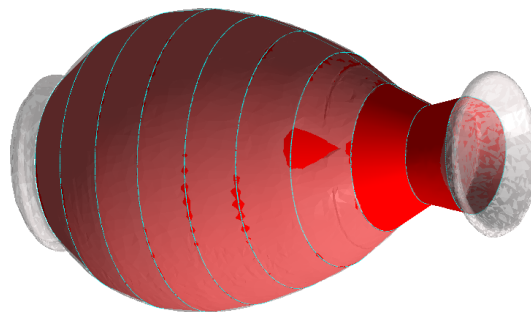


FIGURE 5.5 – Exemple de cylindre généralisé circulaire extrait à partir de sections 2D de *Vase*.

Bien entendu, ces trois exemples sont des cas particuliers de surfaces à forme libre parmi une multitude de possibilités. En effet, il semble difficile de connaître *a priori* quel type de surface a été conçue au départ. De même, une surface extraite de cette manière peut très bien en réalité correspondre à une combinaison de plusieurs surfaces plus petites, modélisées avec des contraintes de continuité. Enfin, les sous-maillages de type inconnu que nous obtenons à la fin de notre processus de rétro-ingénierie peuvent aussi représenter non pas une surface complète mais seulement une partie, initialement conçue par intersection entre une surface à forme libre plus grande et des primitives par exemple.

Dans un premier temps, il semble nécessaire de déterminer le meilleur type de surface (carrés de Bézier, surface NURBS...) et de courbe (courbe de Bézier, B-Spline, NURBS...) en fonction du sous-maillage. Dans un second temps, le problème consiste à approximer correctement cette surface, en fonction de paramètres difficilement automatisables. En effet, il serait intéressant, par exemple, de déduire au préalable des informations sur l'intention de conception originale de la pièce, afin de guider ces différents choix.

Amélioration de l'analyse sémantique des primitives

La modélisation CAO est un domaine très riche avec d'innombrables possibilités de conception pour une pièce manufacturée. Nous souhaitons alors enrichir notre analyse sémantique. Dans un premier temps, il serait intéressant d'étendre notre identification de motifs particuliers à d'autres entités d'usinage et d'autres types de répétitions. Pour cela, nous pourrions lister de nouvelles règles et les ajouter à notre système actuel.

Nous avons par exemple mis en place un ensemble de règles qui permettent d'identifier des congés cylindriques entre deux plans (Fig. 5.6). Par définition, un congé est un cylindre, généralement de faible rayon, qui remplace la droite d'intersection entre les deux plans. Dans ce cas, nous définissons les règles suivantes :

- le cylindre doit être voisin de deux plans qui ne sont pas parallèles;
- le cylindre doit être tangent aux deux plans;
- la droite d'intersection \mathcal{D} entre les deux plans doit être "proche" du cylindre.

Pour la dernière règle, nous construisons tout d'abord le plan de coupe \mathcal{P} contenant les deux droites d'intersection entre le cylindre et chaque plan. Ensuite, nous pouvons utiliser le barycentre du sous-maillage sous-jacent au cylindre, qui en définit seulement une portion. Ainsi, si le barycentre et la droite \mathcal{D} sont du même côté de \mathcal{P} , alors un congé est reconnu (Fig. 5.6.a). Dans l'autre cas, aucun congé n'est reconnu et le cylindre est conservé tel quel (Fig. 5.6.b).

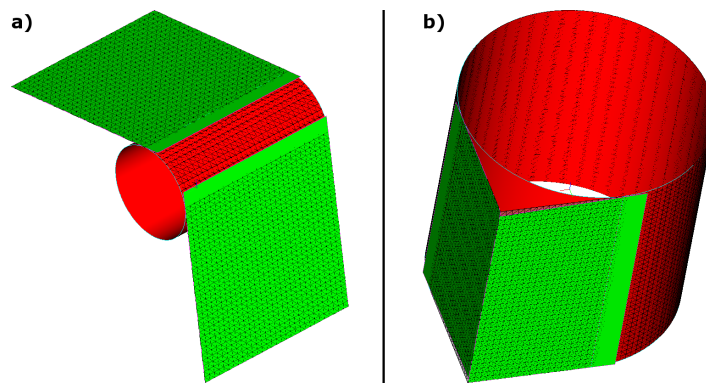


FIGURE 5.6 – Identification de congés cylindriques : a) Cas d'un congé, b) Cas où ce n'est pas un congé.

De même, nous pouvons par exemple extraire les poches simples d'un objet en identifiant un plan voisins de plusieurs primitives par des arêtes concaves (Fig. 5.7). Cependant, des règles supplémentaires sont nécessaires pour identifier correctement les poches plus complexes.

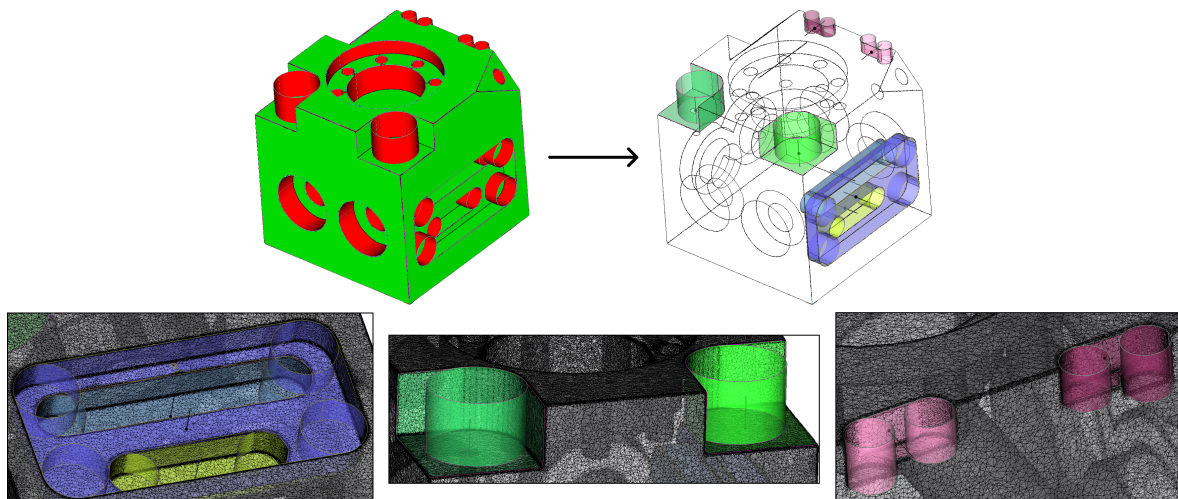


FIGURE 5.7 – Identification de poches simples, ouvertes ou fermées, à partir des primitives de *Block*.

Dans un second temps, il serait intéressant d'analyser d'autres relations géométriques entre les primitives. Par exemple, dans le cas de *Gear*, nous remarquons que les dents de l'engrenage sont inclinées selon des angles approximativement réguliers autour de l'axe principal (Fig. 5.8). Nous pourrions ainsi définir des relations globales et locales, c'est à dire en fonction du repère de référence ou des primitives voisines. Des relations d'égalité dimensionnelle peuvent également être définies, par exemple, entre les largeurs respectives des dents ou leur espacement par rapport à l'axe. Dans ce cas, ces relations pourraient permettre d'identifier une répétition circulaire avec comme entité de référence deux plans formant une dent.

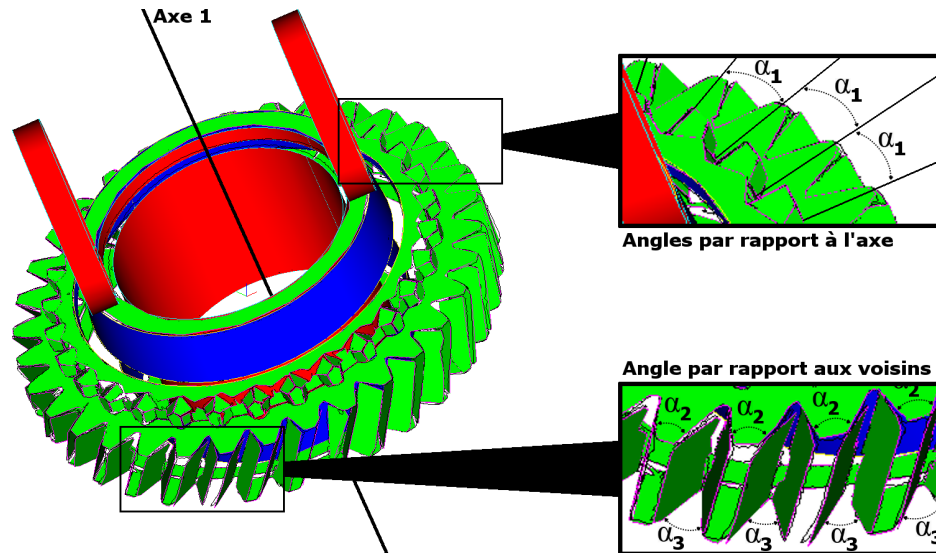


FIGURE 5.8 – Analyse des angles réguliers entre les plans des dents d'engrenage du maillage *Gear*.

Nous souhaiterions également concevoir un algorithme permettant d'ajouter ou de supprimer certaines primitives. En effet, nous avons vu dans le chapitre 3 que certaines primitives pouvaient être incorrectes d'un point de vue conception de l'objet. Les informations sémantiques extraites par notre méthode d'analyse pourraient alors guider un processus de rejet des primitives incohérentes. D'un autre côté, les maillages issus de numérisation peuvent être incomplets à cause d'occultations. Dans ce cas, certaines primitives ne sont pas extraites correctement (Fig 5.9.a). Par conséquent, les répétitions extraites ne sont pas en accord avec le modèle original (Fig 5.9.b). Il serait alors intéressant d'analyser ces répétitions incorrectes pour émettre des hypothèse (Fig 5.9.c) et guider un processus d'extraction supplémentaire (Fig 5.9.d). Cet algorithme permettrait de reconstruire une pièce totalement même si des zones du maillage sont manquantes.

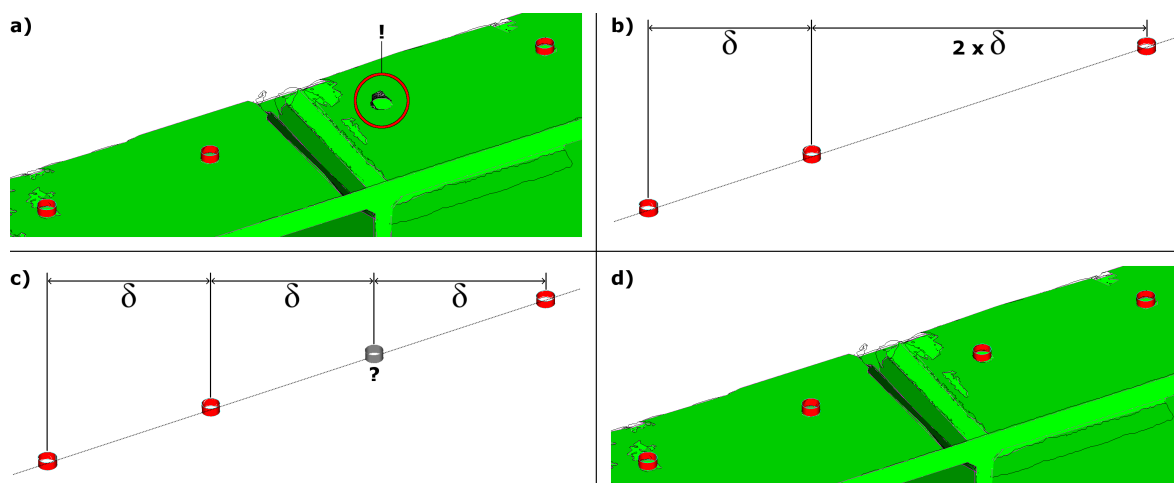


FIGURE 5.9 – Ajout d'une nouvelle primitive en fonction des informations sémantiques : a) Primitives extraites avec un cylindre manquant, b) Identification d'un alignement entre deux des trois cylindres extraits, c) Formulation d'une hypothèse de cylindre manquant, d) Extraction d'un nouveau cylindre.



Liste des publications

Revue internationale

- [1] CAD-Driven Analysis of Reverse Engineered Geometric Models : Application for Beautification
S. Gauthier, G. Subsol, R. Bénière, and W. Puech
IEEE Transaction on Visualization and Computer Graphics ¹
Article soumis en février 2018.

Résumé

Today, it has become more frequent and relatively easy to digitize the surface of 3D objects and then to reconstruct a boundary representation (B-Rep), which is a combination of geometric primitives like planes, cylinders, spheres and cones. However, the obtained results suffer from various inaccuracies, mainly due to noisy data. Moreover, the given reconstruction contains only the geometry, without any information of a semantic nature induced by the design process. In this paper, we present an efficient method to extract absolute references which are not explicitly present in an object, such as the orthogonal coordinate system and alignment planes. These are known as datum references, which are used in the CAD modeling process. Then, we show that datum references can be useful through an application for beautification, which consists in the adjustment of primitive parameters to satisfy relations such as parallelism, orthogonality and concentricity. Our objective is to then design a fast and automatic method, which is seldom seen in reverse engineering. Experimental results applied on reverse engineered 3D meshes show the efficiency and the robustness of our method.

- [2] Analysis of digitized 3D mesh curvature histograms for reverse engineering
S. Gauthier, W. Puech, R. Bénière, and G. Subsol
Computers in Industry ²
Volume 92–93; Pages 67-83; Novembre 2017.

Résumé

Today, it has become more frequent and reasonably easy to digitize the surface of 3D objects. However, the obtained results are often inaccurate and noisy. In this paper, we present an efficient method to analyze a curvature histogram from a digitized 3D surface using a real object. Moreover, we propose to use the curvature histogram analysis for many steps of a reverse engineering process, which can be used to retrieve a CAD model from a digitized one for example. Our objective is to design a fast and fully automated method, which is seldom seen in reverse engineering. Experimental results applied on digitized 3D meshes show the efficiency and the robustness of our proposed method.

1. TVCG : ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=2945
2. CI : www.journals.elsevier.com/computers-in-industry

Conférences internationales

- [3] Orientation beautification of reverse engineered models
S. Gauthier, W. Puech, R. Bénéière, and G. Subsol
GRAPP 2018 : International Conference on Computer Graphics Theory and Applications³
Volume 1 : GRAPP; Pages 91-100; Février 2018; Funchal, Madeira, Portugal.

Résumé

Today, it has become more frequent and relatively easy to digitize the surface of 3D objects and then to reconstruct a boundary representation (B-Rep). However, the obtained results suffer from various inaccuracies, mainly caused by noisy data. In this paper, we present an efficient method to detect and rectify many regularities approximately present in an object. We propose to extract global parallelism and orthogonality constraints, then locally and independently adjust the geometric primitives (planes, cylinders, spheres, cones). We first retrieve orientations from primitives (normals, axes), then compute an orthonormal coordinate system. Finally, we adjust each primitive orientation according to spherical coordinates. Our objective is to design a fast and automatic method, which is seldom seen in reverse engineering. Experimental results applied on reverse engineered 3D meshes show the efficiency and the robustness of our proposed method.

- [4] Digitized 3D mesh segmentation based on curvature analysis
S. Gauthier, W. Puech, R. Bénéière, and G. Subsol
3DIPM 2017 : 3D Image Processing, Measurement, and Applications⁴
Volume : 3DIPM; Pages 33-38; Février 2017; Burlingame, California USA.

Résumé

Today, it is increasingly frequent and easy to digitize the surface of real 3D objects. However, the obtained meshes are often inaccurate and noisy. In this paper, we present a method to segment a digitized 3D surface from a real object by analyzing its curvature. Experimental results - applied on real digitized 3D meshes - show the efficiency of our proposed analysis, particularly in a reverse engineering process.

Conférences nationales

- [5] Segmentation de maillages 3D de pièces manufacturées numérisées : Application à la rétro-conception
S. Gauthier, W. Puech, R. Bénéière, and G. Subsol
CORESA 2016 : COMpression et REprésentation des Signaux Audiovisuels⁵
Mai 2016; Nancy, France.

Distinction : prix du meilleur papier "jeune chercheur"

Résumé

De nos jours, il est de plus en plus fréquent et facile de numériser en 3D la surface des objets réels. Les maillages issus de ces objets numérisés sont souvent imprécis et bruités. De plus, nombre d'entre eux sont de haute résolution, donc très lourds. Dans une chaîne de rétro-conception, bien souvent une étape de segmentation est nécessaire. Dans cet article, nous proposons une nouvelle approche de segmentation pour la rétro-conception utilisant les spécificités des maillages numérisés.

3. GRAPP : www.grapp.visigrapp.org/

4. 3DIPM : www.imaging.org/site/IST/IST/Conferences/EI/EI_2017/Conference/C_3DIPM.aspx

5. CORESA : coresa2016.sciencesconf.org/



Références

- [1] Nabih N Abdelmalek. Segmentation of 3-D range images of small industrial-like objects using the histogram of the surface mean curvature. *International journal of systems science*, 22(10) :1961–1975, 1991. [28](#)
- [2] Marco Attene, Bianca Falcidieno, and Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3) :181–193, 2006. [23](#)
- [3] Bojan Babic, Nenad Nesic, and Zoran Miljkovic. A review of automated feature recognition with rule-based pattern recognition. *Computers in Industry*, 59(4) :321–337, 2008. [47](#), [49](#)
- [4] Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. Learning boundary edges for 3D-mesh segmentation. In *Computer Graphics Forum*, volume 30, pages 2170–2182. Wiley Online Library, 2011. [26](#)
- [5] Roseline Bénière. *Reconstruction d'un modèle B-Rep à partir d'un maillage 3D*. PhD thesis, Université de Montpellier, 2012. [1](#), [5](#), [21](#), [28](#), [65](#), [76](#), [82](#), [87](#), [88](#), [129](#), [130](#)
- [6] Roseline Bénière, Gérard Subsol, Gilles Gesquière, François Le Breton, and William Puech. Recovering primitives in 3D CAD meshes. In *Three-Dimensional Imaging, Interaction, and Measurement*, volume 7864, page 78640R. International Society for Optics and Photonics, 2011. [33](#)
- [7] Roseline Bénière, Gérard Subsol, Gilles Gesquière, François Le Breton, and William Puech. Topology reconstruction for B-Rep modeling from 3D mesh in reverse engineering applications. In *Three-Dimensional Image Processing (3DIP) and Applications II*, volume 8290, page 82900M. International Society for Optics and Photonics, 2012. [42](#)
- [8] Roseline Bénière, Gérard Subsol, Gilles Gesquière, François Le Breton, and William Puech. A comprehensive process of reverse engineering from 3D meshes to CAD models. *Computer-Aided Design*, 45(11) :1382–1393, 2013. [13](#), [14](#), [28](#), [72](#)
- [9] Pal Benko, Géza Kós, Tamás Várady, László Andor, and Ralph Martin. Constrained fitting in reverse engineering. *Computer Aided Geometric Design*, 19(3) :173–205, 2002. [34](#), [57](#)
- [10] Pál Benkő, Ralph R Martin, and Tamás Várady. Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 33(11) :839–851, 2001. [12](#), [32](#), [35](#)
- [11] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, Andrei Sharf, and Claudio Silva. State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports*, volume 1, pages 161–185, 2014. [3](#)
- [12] Fausto Bernardini and Holly Rushmeier. The 3D model acquisition pipeline. In *Computer graphics forum*, volume 21, pages 149–172. Wiley Online Library, 2002. [3](#)

- [13] François Blais. Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1) :231–244, 2004. [3](#)
- [14] Colin Bradley and Bernadette Currie. Advances in the field of reverse engineering. *Computer-Aided Design and Applications*, 2(5) :697–706, 2005. [10](#)
- [15] Marina Bruneau, Alexandre Durupt, Laurent Vallet, Lionel Roucoules, and Jean-Philippe Pernot. A three-level signature by graph for reverse engineering of mechanical assemblies. In *Tools and Methods for Competitive Engineering (Aix-en-Provence : 16 : 2016)*, volume 1, pages 669–681, 2016. [15](#)
- [16] Alan Bundy and Lincoln Wallen. Constructive solid geometry. In *Catalogue of artificial intelligence tools*, pages 21–22. Springer, 1984. [2](#)
- [17] Francesco Buonamici, Monica Carfagni, Rocco Furferi, Lapo Governi, Alessandro Lapini, and Yary Volpe. Reverse engineering modeling methods and tools : a survey. *Computer-Aided Design and Applications*, 15(3) :443–464, 2018. [10](#), [11](#)
- [18] Francesco Buonamici, Monica Carfagni, and Yary Volpe. Recent strategies for 3D reconstruction using reverse engineering : a bird’s eye view. In *Advances on Mechanics, Design Engineering and Manufacturing*, pages 841–850. Springer, 2017. [10](#)
- [19] Francesca Campana and Michele Germani. Datum identification for tolerances control on dense clouds of points. *Computer-Aided Design and Applications*, 5(1-4) :209–219, 2008. [43](#), [44](#)
- [20] Miguel A Carreira-Perpinán. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv :1503.00687*, 2015. [30](#)
- [21] Jack Szu-Shen Chen and Hsi-Yung Feng. Automatic prismatic feature segmentation of scanning-derived meshes utilising mean curvature histograms. *Virtual and Physical Prototyping*, 9(1) :45–61, 2014. [26](#), [29](#)
- [22] Jack Szu-Shen Chen and Hsi-Yung Feng. Idealization of scanning-derived triangle mesh models of prismatic engineering parts. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(2) :205–221, 2017. [14](#), [15](#), [57](#)
- [23] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. In *Acm transactions on graphics (tog)*, volume 28, page 73. ACM, 2009. [16](#)
- [24] Xin Chen and Francis Schmitt. Intrinsic surface properties from surface triangulation. In *European Conference on Computer Vision*, pages 739–743. Springer, 1992. [28](#)
- [25] Jean-Christophe Cuillière, Vincent François, and Rémi Lacroix. A new approach to automatic and a priori mesh adaptation around circular holes for finite element analysis. *Computer-Aided Design*, 77 :18–45, 2016. [1](#)
- [26] Sébastien Delest, Romuald Bone, and Hubert Cardot. Fast segmentation of triangular meshes using waterfall. In *VIIIP*, volume 6, pages 308–312, 2006. [17](#)
- [27] Kris Demarsin, Denis Vanderstraeten, and Dirk Roose. Meshless extraction of closed feature lines using histogram thresholding. *Computer-Aided Design and Applications*, 5(5) :589–600, 2008. [25](#), [29](#)
- [28] Kris Demarsin, Denis Vanderstraeten, Tim Volodine, and Dirk Roose. Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Computer-Aided Design*, 39(4) :276–283, 2007. [25](#)

- [29] William Derigent. *Méthodologie de passage d'un modèle CAO vers un modèle FAO pour des pièces aéronautiques : Prototype logiciel dans le cadre du projet USIQUICK*. PhD thesis, Université Henri Poincaré-Nancy I, 2005. 46
- [30] Luca Di Angelo and Paolo Di Stefano. Geometric segmentation of 3D scanned surfaces. *Computer-Aided Design*, 62 :44–56, 2015. 21
- [31] Andrey Dimitrov, Rongqi Gu, and Mani Golparvar-Fard. Non-uniform B-spline surface fitting from unordered 3D point clouds for as-built modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(7) :483–498, 2016. 130
- [32] Chen-shi Dong and Guo-zhao Wang. Curvatures estimation on triangular mesh. *Journal of Zhejiang University-Science A*, 6(1) :128–136, 2005. 28
- [33] Pierre-Alain Fayolle and Alexander Pasko. An evolutionary approach to the extraction of object construction trees from 3D point clouds. *Computer-Aided Design*, 74 :1–17, 2016. 41
- [34] Martin A Fischler and Robert C Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981. 36, 58
- [35] Robert Fisher, Craig Robertson, and Naoufel Werghi. Constrained object reconstruction incorporating free-form surfaces. In *Proc. IX Spanish Symposium on Pattern Recognition and Image Analysis*, 2001. 130
- [36] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T Silva. Robust moving least-squares fitting with sharp features. *ACM transactions on graphics (TOG)*, 24(3) :544–552, 2005. 33
- [37] Akemi Gálvez and Andrés Iglesias. Firefly algorithm for polynomial Bézier surface parameterization. *Journal of Applied Mathematics*, 2013, 2013. 130
- [38] CH Gao, Frank C Langbein, A David Marshall, and Ralph R Martin. Local topological beautification of reverse engineered models. *Computer-Aided Design*, 36(13) :1337–1355, 2004. 58, 59
- [39] Michael Garland, Andrew Willmott, and Paul S Heckbert. Hierarchical face clustering on polygonal surfaces. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 49–58. ACM, 2001. 23
- [40] Timothy D Gatzke and Cindy M Grimm. Estimating curvature on triangular meshes. *International journal of shape modeling*, 12(01) :1–28, 2006. 28
- [41] Anne Gehre, Isaak Lim, and Leif Kobbelt. Adapting feature curve networks to a prescribed scale. In *Computer Graphics Forum*, volume 35, pages 319–330. Wiley Online Library, 2016. 23
- [42] Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 214–223. ACM, 2004. 19
- [43] Weizhong Geng, Zhengming Chen, Kunjin He, and Yunyan Wu. Feature recognition and volume generation of uncut regions for electrical discharge machining. *Advances in Engineering Software*, 91 :51–62, 2016. 49
- [44] Manish Goyal, Sundar Murugappan, Cecil Piya, William Benjamin, Yi Fang, Min Liu, and Karthik Ramani. Towards locally and globally shape-aware reverse 3D modeling. *Computer-Aided Design*, 44(6) :537–553, 2012. 130

- [45] Azhar Hadmi, William Puech, Brahim Ait Es Said, and Abdellah Ait Ouahman. A robust and secure perceptual hashing system based on a quantization step analysis. *Signal Processing : Image Communication*, 28(8) :929–948, 2013. [28](#)
- [46] JungHyun Han, Mike Pratt, and William C Regli. Manufacturing feature recognition from solid models : a status report. *IEEE transactions on robotics and automation*, 16(6) :782–796, 2000. [47](#), [49](#)
- [47] Ramy Harik. *Spécifications de fonctions pour un système d'aide à la génération automatique de gamme d'usinage : application aux pièces aéronautiques de structure, prototype logiciel dans le cadre du projet RNTL USIQUICK*. PhD thesis, Université Henri Poincaré-Nancy I, 2007. [46](#)
- [48] Jianbing Huang and Chia-Hsiang Menq. Automatic CAD model reconstruction from multiple point clouds for reverse engineering. *Journal of Computing and Information Science in Engineering*, 2(3) :160–170, 2002. [12](#), [13](#)
- [49] Zhi-Kai Huang and Kwok-Wing Chau. A new image thresholding method based on Gaussian mixture model. *Applied Mathematics and Computation*, 205(2) :899–907, 2008. [28](#)
- [50] Cheuk Yiu Ip and William C Regli. A 3D object classifier for discriminating manufacturing processes. *Computers & Graphics*, 30(6) :903–916, 2006. [28](#)
- [51] Sanjay Joshi and Tien-Chien Chang. Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer-Aided Design*, 20(2) :58–66, 1988. [47](#)
- [52] Y Jun, V Raja, and S Park. Geometric feature recognition for reverse engineering using neural networks. *The International Journal of Advanced Manufacturing Technology*, 17(6) :462–470, 2001. [47](#)
- [53] Dong Hwan Kim, Il Dong Yun, and Sang Uk Lee. Boundary-trimmed 3D triangular mesh segmentation based on iterative merging strategy. *Pattern Recognition*, 39(5) :827–838, 2006. [22](#)
- [54] István Kovács, Tamás Várady, and Péter Salvi. Applying geometric constraints for perfecting CAD models in reverse engineering. *Graphical Models*, 82 :44–57, 2015. [45](#), [55](#), [57](#)
- [55] Yu-Kun Lai, Qian-Yi Zhou, Shi-Min Hu, and Ralph R Martin. Feature sensitive mesh segmentation. In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 17–25. ACM, 2006. [19](#)
- [56] Frank C Langbein, A David Marshall, and Ralph R Martin. Numerical methods for beautification of reverse engineered geometric models. In *Geometric Modeling and Processing, 2002. Proceedings*, pages 159–168. IEEE, 2002. [56](#), [57](#)
- [57] Frank Curd Langbein. *Beautification of reverse engineered geometric models*. PhD thesis, Cardiff University, 2003. [56](#)
- [58] Guillaume Lavoué, Florent Dupont, and Atilla Baskurt. A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10) :975–987, 2005. [20](#)
- [59] Truc Le and Ye Duan. A primitive-based 3D segmentation algorithm for mechanical CAD models. *Computer Aided Geometric Design*, 52 :231–246, 2017. [44](#), [45](#)
- [60] Kai Wah Lee and Pengbo Bo. Feature curve extraction from point clouds via developable strip intersection. *Journal of Computational Design and Engineering*, 3(2) :102–111, 2016. [24](#)

- [61] Ke Li, Gilles Foucault, J-C Léon, and Moreno Trlin. Fast global and partial reflective symmetry analyses using boundary surfaces of mechanical components. *Computer-Aided Design*, 53 :70–89, 2014. [54](#), [55](#)
- [62] Ming Li, Frank C Langbein, and Ralph R Martin. Detecting design intent in approximate CAD models using symmetry. *Computer-Aided Design*, 42(3) :183–201, 2010. [54](#)
- [63] Yangyan Li, Xiaokun Wu, Yiorgos Chrysathou, Andrei Sharf, Daniel Cohen-Or, and Niloy J Mitra. Globfit : Consistently fitting primitives by discovering global relations. In *ACM Transactions on Graphics (TOG)*, volume 30, page 52. ACM, 2011. [37](#), [58](#)
- [64] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2) :451–461, 2003. [18](#)
- [65] Katia Lupinetti, Franca Giannini, Marina Monti, and Jean-Philippe Pernot. Identification of functional components in mechanical assemblies. *Procedia CIRP*, 60 :542–547, 2017. [48](#)
- [66] David A Madsen and David P Madsen. *Geometric Dimensioning and Tolerancing*. Goodheart-Willcox Co., 2012. [43](#)
- [67] Evgeni Magid, Octavian Soldea, and Ehud Rivlin. A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data. *Computer Vision and Image Understanding*, 107(3) :139–159, 2007. [28](#)
- [68] Atef Masmoudi, Slim Chaoui, and Afif Masmoudi. A finite mixture model of geometric distributions for lossless image compression. *Signal, Image and Video Processing*, 10(4) :671–678, 2016. [30](#)
- [69] Atef Masmoudi, Afif Masmoudi, and William Puech. A new semiparametric finite mixture model-based adaptive arithmetic coding for lossless image compression. *Circuits, Systems, and Signal Processing*, 35(4) :1163–1186, 2016. [28](#)
- [70] Hiroshi Masuda, Ichiro Tanaka, and Masakazu Enomoto. Reliable surface extraction from point-clouds using scanner-dependent parameters. *Computer-Aided Design and Applications*, 10(2) :265–277, 2013. [14](#), [66](#)
- [71] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007. [30](#)
- [72] Kristýna Micháľková and Bohumir Bastl. Imposing angle boundary conditions on B-spline/NURBS surfaces. *Computer-Aided Design*, 62 :1–9, 2015. [130](#)
- [73] Niloy J Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3D geometry : Extraction and applications. In *Computer Graphics Forum*, volume 32, pages 1–23. Wiley Online Library, 2013. [47](#), [53](#), [54](#)
- [74] Tomohiro Mizoguchi, Hiroaki Date, Satoshi Kanai, and Takeshi Kishinami. Segmentation of scanned mesh into analytic surfaces based on robust curvature estimation and region growing. In *International Conference on Geometric Modeling and Processing*, pages 644–654. Springer, 2006. [20](#), [21](#)
- [75] Prashant Mohan, Payam Haghighi, Jami J Shah, and Joseph K Davidson. Automatic detection of directions of dimensional control in mechanical parts. In *ASME 2014 International Manufacturing Science and Engineering Conference collocated with the JSME 2014 International Conference on Materials and Processing and the 42nd North American Manufacturing Research Conference*, pages V001T04A005–V001T04A005. American Society of Mechanical Engineers, 2014. [44](#)

- [76] Jorge J Moré. The Levenberg-Marquardt algorithm : implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978. [34](#), [77](#)
- [77] Adel Moussaoui and Samy Ait-Aoudia. Generator of 2D geometric constraint graphs. *Computer-Aided Design and Applications*, 13(3) :271–280, 2016. [103](#)
- [78] Lakshmi Priya Muraleedharan, Shyam Sundar Kannan, Ameya Karve, and Ramanathan Muthuganapathy. Random cutting plane approach for identifying volumetric features in a CAD mesh model. *Computers & Graphics*, 70 :51–61, 2018. [50](#)
- [79] Zhibin Niu, Ralph R Martin, Frank C Langbein, and Malcolm A Sabin. Rapidly finding CAD features using database optimization. *Computer-Aided Design*, 69 :35–50, 2015. [50](#)
- [80] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, volume 35, pages 203–215. Wiley Online Library, 2016. [35](#), [58](#)
- [81] Nursel Öztürk and Ferruh Öztürk. Neural network based non-standard feature recognition to integrate CAD and CAM. *Computers in Industry*, 45(2) :123–135, 2001. [47](#)
- [82] Pralay Pal and Anjani Kumar. A hybrid approach for identification of 3D features from CAD database for manufacturing support. *International Journal of Machine Tools and Manufacture*, 42(2) :221–228, 2002. [47](#)
- [83] Pralay Pal, AM Tigga, and A Kumar. Feature extraction from large CAD databases using genetic algorithm. *Computer-Aided Design*, 37(5) :545–558, 2005. [47](#)
- [84] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3) :1065–1076, 1962. [31](#)
- [85] Mark Pauly, Niloy J Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J Guibas. Discovering structural regularity in 3D geometry. *ACM transactions on graphics (TOG)*, 27(3) :43, 2008. [52](#)
- [86] Sylvain Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys (CSUR)*, 34(2) :211–262, 2002. [10](#)
- [87] V Rameshbabu and MS Shunmugam. Hybrid feature recognition method for setup planning from STEP AP-203. *Robotics and Computer-Integrated Manufacturing*, 25(2) :393–408, 2009. [47](#)
- [88] Medhat Rashad, Mohamed Khamiss, and Mohamed MOUSA. A review on mesh segmentation techniques. *International Journal of Engineering and Innovative Technology (IJEIT)*, 6(8) :18–26, 2017. [17](#)
- [89] Rui SV Rodrigues, José FM Morgado, and Abel JP Gomes. Part-based mesh segmentation : A survey. In *Computer Graphics Forum*. Wiley Online Library, 2018. [16](#), [17](#)
- [90] Pedro V Sander, John Snyder, Steven J Gortler, and Hugues Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 409–416. ACM, 2001. [16](#)
- [91] Emilio M Sanfilippo and Stefano Borgo. What are features? An ontology-based review of the literature. *Computer-Aided Design*, 80 :9–18, 2016. [46](#)
- [92] Giovanna Sansoni, Marco Trebeschi, and Franco Docchio. State-of-the-art and applications of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9(1) :568–601, 2009. [1](#)

- [93] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007. [37](#)
- [94] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2) :461–464, 1978. [30](#)
- [95] Jami J Shah. Conceptual development of form features and feature modelers. *Research in engineering design*, 2(2) :93–108, 1991. [46](#)
- [96] Craig M Shakarji. Least-squares fitting algorithms of the NIST algorithm testing system. *Journal of research of the National Institute of Standards and Technology*, 103(6) :633, 1998. [34](#), [77](#), [103](#)
- [97] Ariel Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008. [16](#)
- [98] Oana Sidi, Oliver van Kaick, Yanir Kleiman, Hao Zhang, and Daniel Cohen-Or. *Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering*, volume 30. ACM, 2011. [16](#)
- [99] Jan A Snyman. *Practical mathematical optimization*. Springer, 2005. [77](#)
- [100] Nikhil Somani, Markus Rickert, and Alois Knoll. An exact solver for geometric constraints with inequalities. *IEEE Robotics and Automation Letters*, 2(2) :1148–1155, 2017. [103](#)
- [101] Ian Stroud. *Boundary representation modelling techniques*. Springer Science & Business Media, 2006. [2](#)
- [102] VB Sunil, Rupal Agarwal, and SS Pande. An approach to recognize interacting features from B-Rep CAD models of prismatic machined parts using a hybrid (graph and rule based) technique. *Computers in Industry*, 61(7) :686–701, 2010. [48](#), [50](#)
- [103] Songqiao Tao, Zhengdong Huang, Lujie Ma, Shunsheng Guo, Shuting Wang, and Youbai Xie. Partial retrieval of CAD models based on local surface region decomposition. *Computer-Aided Design*, 45(11) :1239–1252, 2013. [47](#)
- [104] Panagiotis Theologou, Ioannis Pratikakis, and Theoharis Theoharis. A comprehensive overview of methodologies and performance evaluation frameworks in 3D mesh segmentation. *Computer Vision and Image Understanding*, 135 :49–82, 2015. [16](#)
- [105] Grit Thürrner and Charles A Wüthrich. Computing vertex normals from polygonal facets. *Journal of Graphics Tools*, 3(1) :43–46, 1998. [27](#)
- [106] Zahra Toony, Denis Laurendeau, and Christian Gagné. Describing 3D geometric primitives using the Gaussian sphere and the Gaussian accumulator. *3D Research*, 6(4) :42, 2015. [32](#), [33](#)
- [107] Trung-Thien Tran, Van-Toan Cao, and Denis Laurendeau. Extraction of reliable primitives from unorganized point clouds. *3D Research*, 6(4) :44, 2015. [36](#)
- [108] RJ Urbanic and WH Elmaraghy. Design recovery of internal and external features for mechanical components. *Virtual and Physical Prototyping*, 3(2) :61–83, 2008. [46](#), [52](#), [53](#)
- [109] Tamás Várady, Michael A Facello, and Zsolt Terék. Automatic extraction of surface structures in digital shape reconstruction. *Computer-Aided Design*, 39(5) :379–388, 2007. [24](#)

- [110] Harold Vilmart, Jean-Claude Léon, and Federico Ulliana. From CAD assemblies toward knowledge-based assemblies using an intrinsic knowledge-based assembly model. *Computer-Aided Design and Applications*, 15(3) :300–317, 2018. [50](#), [51](#), [52](#), [55](#), [56](#)
- [111] Hao Wang, Tong Lu, Oscar Kin-Chung Au, and Chiew-Lan Tai. Spectral 3D mesh segmentation with a novel single segmentation field. *Graphical models*, 76(5) :440–456, 2014. [16](#)
- [112] Jun Wang, Dongxiao Gu, Zhanheng Gao, Zeyun Yu, Changbai Tan, and Laishui Zhou. Feature-based solid model reconstruction. *Journal of Computing and Information Science in Engineering*, 13(1) :011004, 2013. [35](#)
- [113] Jun Wang, Dongxiao Gu, Zeyun Yu, Changbai Tan, and Laishui Zhou. A framework for 3D model reconstruction in reverse engineering. *Computers & Industrial Engineering*, 63(4) :1189–1200, 2012. [13](#), [41](#)
- [114] Qingmai Wang and Xinghuo Yu. Ontology based automatic feature recognition framework. *Computers in Industry*, 65(7) :1041–1052, 2014. [50](#), [51](#)
- [115] Kouki Watanabe and Alexander G Belyaev. Detection of salient curvature features on polygonal surfaces. In *Computer Graphics Forum*, volume 20, pages 385–392. Wiley Online Library, 2001. [23](#)
- [116] Marco Weißgerber, Marko Ebermann, Sophie Gröger, and Erhard Leidich. Requirements for datum systems in computer aided tolerancing and the verification process. *Procedia CIRP*, 43 :238–243, 2016. [46](#)
- [117] Kaufui V Wong and Aldo Hernandez. A review of additive manufacturing. *ISRN Mechanical Engineering*, 2012, 2012. [1](#)
- [118] Yanyan Wu, Jami J Shah, and Joseph K Davidson. Computer modeling of geometric variations in mechanical parts and assemblies. *Journal of Computing and Information Science in Engineering*, 3(1) :54–63, 2003. [47](#)
- [119] Yuguang Wu and Qiqi Gu. The composition principle of the datum reference frame. *Procedia CIRP*, 43 :226–231, 2016. [44](#)
- [120] Xiang Yang, Peter Meer, and Hae Chang Gea. Robust recovery of 3D geometric primitives from point cloud. In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages V001T02A009–V001T02A009. American Society of Mechanical Engineers, 2017. [37](#)
- [121] Bing Yi, Zhenyu Liu, Jianrong Tan, Fengbei Cheng, Guifang Duan, and Ligang Liu. Shape recognition of CAD models via iterative slippage analysis. *Computer-Aided Design*, 55 :13–25, 2014. [19](#)
- [122] Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. Fast and robust detection of crest lines on meshes. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 227–232. ACM, 2005. [23](#)
- [123] Xue-Feng Zhu, Ping Hu, Zheng-Dong Ma, Xiangkui Zhang, Weidong Li, Jingru Bao, and Mingzeng Liu. A new surface parameterization method based on one-step inverse forming for isogeometric analysis-suited geometry. *The International Journal of Advanced Manufacturing Technology*, 65(9-12) :1215–1227, 2013. [130](#)