

Université de Montréal

**Reconstruction polyédrique de scènes en trois dimensions à partir de
cartes de profondeurs**

par
Valentin Vial

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Avril, 2007

© Valentin Vial, 2007.



QA

76

U54

2007

V.033

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

**Reconstruction polyédrique de scènes en trois dimensions à partir de
cartes de profondeurs**

présenté par:

Valentin Vial

a été évalué par un jury composé des personnes suivantes:

Pierre Poulin
président-rapporteur

Max Mignotte
directeur de recherche

Sébastien Roy
membre du jury

Mémoire accepté le 20 juillet 2007

RÉSUMÉ

Nous allons présenter dans ce mémoire une méthode de reconstruction en trois dimensions qui se base sur des cartes de profondeurs de la forme que l'on cherche à reconstruire, prises sous différents angles de vue. Pour chacune des directions, une carte de profondeurs nous donne "l'épaisseur" de la forme sur un quadrillage de la vue observée. Ces données sont donc de type tomographique. Pour chaque angle de vue, ces cartes sont obtenues à partir de données visuelles observées à l'avant et à l'arrière de la forme étudiée où l'on relève des distances et qui permettent ainsi d'en déduire les profondeurs. Les occlusions n'apparaissent donc pas dans ces données. On réalise ensuite les mêmes projections sur une forme polyédrique arbitraire de départ que l'on va chercher à déformer de façon à obtenir l'objet visé. Cette déformation se fait de façon récursive en cherchant à minimiser, grâce à une descente de gradient, une fonction d'énergie qui calcule la différence entre les profondeurs des données de la forme étudiée et celles de la forme que l'on modèle.

Mots clés: tomographie, carte de profondeurs, reconstruction en trois dimensions, descente de gradient, polyèdre

ABSTRACT

We present in this thesis a method of reconstruction in three dimensions which is based on maps of depths of the form that we want to reconstruct, taken under various angles of sight. For each direction, a map of depths gives us “the thickness” of the form on a grid of the sight observed. This data is thus of a tomographic type. For each angle of sight, these maps are obtained starting from visual data observed at the front and the back of the form observed, where we measure the distances and which thus makes it possible to deduce depths from them. Occlusions do not appear in this data. We then do the same projections of an arbitrary polyhedral original shape that we seek to deform in order to obtain the object to be reconstructed. This deformation is made recursively while seeking to minimize, with a gradient descent, an energy function which computes the differences between the depths of the data of the studied shape and those of the shape which we model.

Keywords: tomography, map of depth, three dimensional reconstruction, gradient descent, polyhedron

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES FIGURES	vi
NOTATION	vii
REMERCIEMENTS	viii
CHAPITRE 1 : INTRODUCTION	1
CHAPITRE 2 : MÉTHODE DE RECONSTRUCTION EN TROIS DIMENSIONS	2
2.1 Introduction	2
2.2 Stéréoscopie	3
2.2.1 Méthodes locales de base	4
2.2.2 Méthodes globales	7
2.2.3 Méthodes dynamiques	7
2.3 Méthode de coloriage de voxels	9
2.4 Méthode de reconstruction basée sur de forts <i>a priori</i>	13
CHAPITRE 3 : RECONSTRUCTION POLYÉDRIQUE À PARTIR DE CARTES DE PROFONDEURS	19
3.1 Principe	19
3.2 Résumé de l'algorithme principal	20

3.3	Principe de la tomographie	21
3.4	Obtention des cartes de profondeurs	24
3.5	Algorithme principal	26
3.5.1	Définition des lignes de "projection"	26
3.5.2	Initialisation de la forme de départ	27
3.5.3	Expression du gradient de la fonction d'énergie	34
3.5.4	Calcul des valeurs des cartes de projection : $p_{\theta,i,j}(v)$	35
3.5.5	Calcul de $\frac{\partial p_{\theta,i,j}}{\partial v_i}(v)$ présent dans l'expression du gradient	38
3.5.5.1	Restriction du volume à considérer	39
3.5.5.2	Découpage en volumes élémentaires	41
3.5.5.2.1	Calcul d'un point supplémentaire : e	41
3.5.5.3	Définition de trois cas à distinguer pour les calculs	45
3.5.5.4	Découpage en sous-volumes élémentaires	46
3.5.5.5	Calcul des $p(i, j)$	47
3.5.5.6	Calcul de la hauteur des axes : P	53
3.5.5.7	Calcul des $\frac{\partial p}{\partial v_i}(i, j)$	57
3.5.5.7.1	Calcul des $\frac{\partial P}{\partial v_0}$	58
3.5.5.7.2	Calcul des $\frac{\partial p}{\partial v_i}(i, j)$	62
3.5.6	Résumé de l'algorithme	66
CHAPITRE 4 : RÉSULTATS EXPÉRIMENTAUX.		68
4.1	Présentation des conditions des expériences	68
4.2	Tests basiques	73
4.3	Tests avec des formes plus évoluées	76
4.4	Mise en évidence de l'importance du choix des vues et du problème des occlusions	82
4.5	Tests sur des formes complexes	93

4.6 Temps de calcul 100

CHAPITRE 5 : DISCUSSION ET CONCLUSION 101

BIBLIOGRAPHIE 104

LISTE DES FIGURES

2.1	Schéma représentant la géométrie de l'expérience. On voit apparaître les deux plans images définis par les vecteurs u_1, v_1 et u_2, v_2 . Le plan équipolaire Π , et les deux droites mises en correspondance l_{m1} et l_{m2} .	4
2.2	Illustration d'un point de l'image 1 projeté dans le monde réel à plusieurs profondeurs et reprojété sur l'image 2.	6
2.3	Exemple d'un graphe pour minimiser l'énergie d'une droite de trois pixels avec deux profondeurs possibles à chaque fois. A gauche, on fait apparaître les cases du tableau correspondant au coût. Au milieu le graphe associé où les arcs horizontaux correspondent à des coûts de lissage. A droite un exemple de coupure minimale. Les profondeurs retenues pour chaque pixel correspondent aux arcs verticaux coupés.	9
2.4	Schéma illustrant le cas d'une petite et d'une grande "baseline". . .	10
2.5	Coloriage de voxels dans le cas le plus simple. La forme étudiée est le disque orange et la forme obtenue est la partie grisée. Celle-ci est approximative et n'est acceptable que lorsque la forme est convexe et en deux dimensions.	12
2.6	Exemple de résultats obtenus par coloriage de voxels avec contrainte sur les caméras.	13
2.7	Schéma d'un volume que l'on "creuse" par la méthode du "space carving".	14
2.8	Exemple de résultat obtenu par "space carving"	15
2.9	Exemple de reconstruction obtenue à partir de deux radiographies. .	17

- 2.10 Exemple de reconstruction de cage thoracique. (a) et (b) sont les radiographies sur lesquelles on travaille. (c) et (d) sont les résultats retenus. 18
- 3.1 On fait ici apparaître un point v_i de la forme avec ses voisins (figure de gauche). On déplace ensuite celui-ci et on observe que seul le volume défini par le point v_i et ses voisins (représenté en gras) est modifié. Le reste de la forme est inchangée. Le calcul de dérivation ne se fera donc qu'en étudiant ce volume. 22
- 3.2 Un triangle isocèle de base d et de hauteur h et un losange de diagonale d et h . Projeté sur une direction verticale, ces deux formes engendrent les même données tomographiques. 23
- 3.3 Sur la forme C_1 , la projection le long de $d1$ est celle obtenue par un rayon, la valeur rapportée sera donc $l_1 + l_2$. Dans le cas d'une mesure visuelle, forme C_2 , seuls les premiers obstacles au-dessus et en-dessous sont détectés, on ignore ce qu'il y a entre. La valeur enregistrée est alors l_3 , celle-ci est fausse et sera de plus en contradiction avec des projections faites dans d'autres directions. Par exemple une projection horizontale enregistrera une valeur l_4 alors que la mesure verticale l_3 sous-entend que l'on devrait trouver la valeur l_5 (ce que l'on trouverait si la pièce était pleine). 25
- 3.4 Illustration des lignes de projection d'équation $x = i$ et $y = j$ à travers le volume V . Par soucis de visibilité, on n'a représenté qu'une partie de ces lignes. 29
- 3.5 Schéma illustrant les données calculées, on pourra par abus de langage parler de données de projection même si il s'agit d'intégration le long de droite dans l'espace. 30

3.6	On voit ici la « sphère » représentée par quatre points : v_1, v_2, v_3 et v_4 inscrite dans un cube.	31
3.7	Illustration de la division d'une facette triangulaire en quatre autres facettes triangulaires.	32
3.8	Sphères obtenues après une, deux et trois itérations.	33
3.9	On voit ici l'intersection m_1 de la droite de projection d_1 avec le plan P_f formé par la facette $v_0v_1v_2$. m_1 n'appartient pas au triangle $v_0v_1v_2$ donc il sera ignoré. La droite d_2 coupe le plan P_f en m_2 qui appartient au triangle. On enregistrera donc la hauteur de ce point.	36
3.10	Représentation du triangle $p_0p_1p_2$ et d'un point M à l'intérieur de celui-ci. La zone en pointillés forts correspond à l'ensemble des points qui vérifie $p_0\vec{p}_1 \cdot p_0\vec{M} \cdot p_0\vec{p}_2 \cdot p_0\vec{M} > 0$. La deuxième zone en pointillés faibles correspond à la zone où $p_1\vec{p}_2 \cdot p_1\vec{M} \cdot p_1\vec{p}_0 \cdot p_1\vec{M} > 0$	38
3.11	Exemple en 2 dimensions d'une droite $l_{i,j}$ coupant la figure en deux hauteurs z_0 et z_1 . n_{i-1} est orienté vers le haut et n_i vers le bas donc on aura $p_{i,j}(v) = z_0 - z_1$ ce qui correspond bien à la distance traversée.	39
3.12	Volume formé par le point v_i et ses voisins que l'on isole du reste de la forme.	40
3.13	Découpage d'un volume défini avec les voisins du point v_i en 3 volumes élémentaires vo_0, vo_1 et vo_2 . (Soit encore les volumes formés par les points $v_i v_{i0} v_{i3} v_{i4}, v_i v_{i0} v_{i1} v_{i2}$ et $v_i v_{i0} v_{i2} v_{i3}$.)	42
3.14	Ajout d'un point e . On crée ici les volumes : $v_i v_{i0} e v_{i1}, v_i v_{i1} e v_{i2}, v_i v_{i2} e v_{i3}, v_i v_{i3} e v_{i4}$ et $v_i v_{i4} e v_{i0}$	44
3.15	Volume élémentaire constitué du point v_i , de deux de ses voisins v_{i1} et v_{i2} et du point e	45

3.16	Illustration des trois cas en fonction de la projection du volume élémentaire suivant l'axe (O, z)	46
3.17	On fait apparaître sur cette figure un déplacement de v_i qui entraîne ainsi une variation d_l de la distance l traversée par la droite $l_{i,j}$ à travers le volume.	47
3.18	Schéma représentant le volume d'origine formé des points p_0, p'_0, p'_1 et p'_2 et celui modifié avec les points p_0, p'_0, p_1 et p_2 . On fait apparaître les points d'intersections entre la droite de projection et le volume d'origine, soit : r'_1 et r'_2 et ceux avec le volume modifié : r_1 et r_2 . . .	50
3.19	Figure illustrant l'invariance de la longueur DE pour des d, h et b constants.	51
3.20	Schéma d'un volume élémentaire (dans le cas où sa projection forme un triangle) divisé en trois volumes avec l'ajout du point p'_0	53
3.21	Schéma d'un volume élémentaire (dans le cas où sa projection forme un quadrilatère) divisé en quatre volumes avec l'ajout des points p'_4 et p''_4	55
3.22	Volume élémentaire sur lequel on effectue les calculs.	59
4.1	Une forme représentée sous OpenGL vue de profil, puis vue après une rotation de 180 degrés. La correspondance des points de ces deux vues avec leurs profondeurs permet de constituer une carte de profondeurs.	69
4.2	Première image : Schématisation d'une carte de profondeurs. Les parties les plus noires sont les plus épaisses et inversement pour les parties les plus blanches. Deuxième image : voiture d'origine en vue oblique.	70
4.3	sphère vue 90 0 itérations 300 pas 0.01	73

4.4	Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue. Nous rappelons que cette figure, ainsi que les suivantes sont contenues dans des fenêtres carrées de cent millimètres de côté avec une marge d'environ cinq à dix millimètres sur les bords.	74
4.5	Cube, vue 90 0 0 itérations 300 pas 0.01	75
4.6	10 itérations, 50 itérations, 100 itérations et 150 itérations	77
4.7	Lampe, vues 90 0 0 90 itérations 200 pas 0.01	78
4.8	Représentation de la courbe de la fonction d'énergie calculée au cours des itérations.	79
4.9	Diagramme de répartition d'erreurs. Superposition de l'image de référence et de l'image obtenue selon deux points de vue.	80
4.10	Superposition de l'image de référence et de l'image obtenue. Cartographie de l'erreur : les couleurs extrêmes rouge ou bleu foncée indiquent de fortes erreurs, celles entre le vert et le jaune indiquent des erreurs plus faibles.	81
4.11	Cylindre-sphère vues 90 0 0 méthode 0 itérations 200 pas 0.01	83
4.12	Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.	84
4.13	Arche vues 90 0 0 90 méthode 0 itérations 200 pas 0.01	85
4.14	Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.	86
4.15	Arche vues 90 0 0 90 méthode 0 itérations 200 pas 0.01	87
4.16	Images du haut : vue 0 0, énergie : 0.25, images du bas : vue 0 0 90 0, énergie : 1.60	89

4.17	Images du haut : vue : 0 0 90 0 0 90, énergie : 2.05, images du bas :	
	vue : 0 0 90 0 0 90 45 0, énergie : 8.1	90
4.18	Images du haut : vue 0 0 90 0 0 90 45 0 0 45, énergie : 9.4, images du	
	bas : vue : 0 0 90 0 0 90 45 0 0 45 20 40 30 20 80 15 30 10, énergie : 30	91
4.19	Voiture : forme cible	93
4.20	Voiture : vue de profil	94
4.21	Voiture : vue de face	94
4.22	Voiture : vue oblique Voiture vues 90 0 0 90 méthode 0 itérations	
	200 pas 0.01	95
4.23	Diagramme de répartition d'erreurs et superposition de l'image de	
	référence et de l'image obtenue.	96
4.24	Corps vues 90 0 0 90 méthode 0 itérations 200 pas 0.01	98
4.25	Diagramme de répartition d'erreurs et superposition de l'image de	
	référence et de l'image obtenue.	99
4.26	Durée de calcul pour effectuer 200 itérations avec un nombre de vues	
	et de faces variable réalisé avec un AMD Athlon XP 2800+.	100

NOTATION

On note v ou \vec{v} suivant le contexte pour désigner l'ensemble des points constituant la forme que l'on modèle pour retrouver la forme cible.

v_i représente donc le i -ème point de la forme.

d l'ensemble des valeurs des cartes de projection de la forme étudiée que l'on cherche à reconstruire.

$d_{\theta,i,j}$ représente la valeur de la carte de profondeur prise sous l'angle θ au niveau de la droite d'équation $x = i, y = j$.

p et $p_{\theta,i,j}$ représentent les mêmes données mais pour la forme que l'on modélise. On pourra aussi les noter : $p(v)$ et $p_{\theta,i,j}(v)$ pour préciser que ces données dépendent des points v de la forme.

Pour un point d'indice i , on utilisera les notations x_i, y_i et z_i pour désigner ses coordonnées.

$\frac{\partial}{\partial v_i}$ représente la dérivé partielle par rapport au point v_i . \vec{i}, \vec{j} et \vec{k} représentent les vecteurs unitaires selon les axes respectifs $(O, x), (O, y)$ et (O, z) . Tous les repères dans lesquels nous travaillons sont directs et orthonormés. On a ainsi :

$$\frac{\partial}{\partial v_i} = \frac{\partial}{\partial x_i} \vec{i} + \frac{\partial}{\partial y_i} \vec{j} + \frac{\partial}{\partial z_i} \vec{k}.$$

REMERCIEMENTS

Je remercie tout d'abord mon directeur de recherche Max Mignotte qui m'a proposé ce sujet et qui m'a beaucoup aidé à la réalisation de ce présent mémoire. Je remercie également Said Benameur pour le temps qu'il m'a consacré et le logiciel qu'il a mis à ma disposition pour effectuer les calculs d'erreur. Je remercie les membres de mon jury Sébastien Roy et Pierre Poulin pour l'étude de ce document.

Un grand merci à mes parents pour leur soutien et la relecture de mon mémoire. Merci à mes grands parents, à mon frère et mes soeurs. Merci aux gousus. Enfin un remerciement très spécial à toutes les personnes qui m'ont cotoyé et encouragé durant ma maîtrise. Je commence tout d'abord par le directeur des directeurs : Pierre. Merci également au Pédro, à la Madelailene, au Cyril, au Vince mon directeur adjoint, au JB mon ancien colocataire, à la Mao, à l'Anna, à Fabinou, à l'Astro, au Tapir, à la Rouste, au Grossins, à l'Alban, au Laurent, à l'Olive directeur des réunions, au Frouss, au Ben, à la Lydie. Merci à tous les Ibervillois et autres amis que je n'ai pas cités.

CHAPITRE 1

INTRODUCTION

Le sujet de ce mémoire tourne autour du champ d'étude de la reconstruction en trois dimensions. Ce domaine est très vaste et très développé notamment en vision par ordinateur et en robotique. De nombreuses méthodes ont été conçues à ce sujet. On peut distinguer parmi la plupart d'entre elles quelques grands axes de méthodes de base qui ont été ensuite améliorées et complexifiées au travers de différentes études. On observe ensuite également des algorithmes très spécifiques à leur domaine d'application, ce qui les place très à part des autres mais qui permettent de très bonnes optimisations de la reconstruction. Enfin, deux points importants dans toute méthode de reconstruction en trois dimensions sont la nature des données à partir desquelles on travaille, et le procédé mis en œuvre pour obtenir celles-ci. Dans notre cas, nous repartirons d'une méthode de reconstruction purement mathématique qui se base sur le traitement de cartes de profondeurs prises sous différents angles de vue pour reconstituer la forme visée. Nous nous distinguerons du programme original sur la nature de ces cartes et la façon de les obtenir. Nous verrons dans une première partie un état de l'art dans le domaine de la reconstruction en trois dimensions en présentant les méthodes générales de ce domaine. Puis nous verrons notre propre méthode en la développant étape par étape et en faisant ressortir les aspects qui la distinguent des autres. Nous présenterons après cela une série de résultats illustrant les forces et les faiblesses de notre algorithme. Enfin nous terminerons par une discussion et une conclusion.

CHAPITRE 2

MÉTHODE DE RECONSTRUCTION EN TROIS DIMENSIONS

2.1 Introduction

Le domaine de la reconstruction en trois dimensions est un sujet très étudié dans le secteur de l'imagerie. Celui-ci est depuis cette dernière décennie en constante évolution. Il existe donc aujourd'hui un grand nombre de méthodes et d'algorithmes traitant de ce problème. Ces derniers sont variés et exploitent toutes sortes d'idées. Tous ont cependant leurs avantages et leurs défauts. Leur utilisation dépend en effet des contraintes matérielles : qualité et précision des caméras, appareils photo, support pour les positionner, logistique de l'acquisition des données ; des contraintes de l'objet ou de la scène étudiée ainsi que celles concernant l'environnement : dimension de la scène ou de l'objet, distances intervenant entre les objectifs et les sujets, nature de la scène complexe ou simple, statique ou en mouvement, nature des lumières et des réflexions des objets, présence d'occlusions c'est-à-dire de points visibles seulement à partir de certains points de vue ; des exigences recherchées : qualité et précision, temps de calcul ; et enfin des pré-requis et de la part d'action de l'utilisateur : connaissance approximative de la forme recherchée, automatisation de la reconstruction ou intervention de l'utilisateur au cours de certaines étapes.

On comprend donc qu'il est difficile de faire le tour de l'ensemble de ces applications. On peut néanmoins observer que la plus grande partie des méthodes générales tournent autour du principe de la stéréoscopie. C'est ce sujet que nous développerons dans un premier temps. Nous étudierons ensuite plus précisément l'un de ces algorithmes les plus connus en reconstruction appelé « space carving

» ou « creusage de l'espace ». Nous verrons ensuite un exemple de méthode spécialisée dans le domaine médical et ayant la particularité de se baser sur de fortes connaissances *a priori*. Enfin nous porterons notre réflexion de façon précise sur notre méthode qui repose sur le traitement de cartes de profondeurs pour effectuer la reconstruction en trois dimensions du sujet étudié.

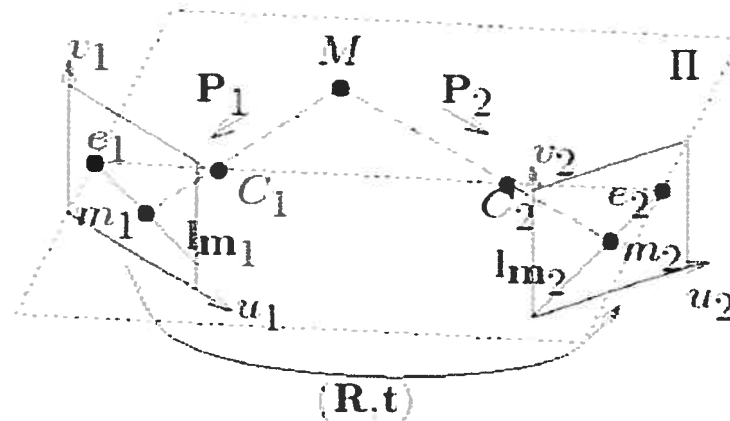
2.2 Stéréoscopie

La stéréoscopie constitue l'un des axes majeurs de la reconstruction en trois dimensions. C'est également l'une des méthodes les plus anciennes. Elle se base sur le même principe de fonctionnement que la vue humaine. Il s'agit de retrouver les profondeurs des objets d'une scène à partir de deux images de cette même scène mais prises à des points de vue différents.

Dans le cas général et simplifié, on doit tout d'abord faire certaines hypothèses sur la scène étudiée et sur le matériel utilisé. On doit tout d'abord faire l'hypothèse que les surfaces des objets de la scène sont de type lambertiennes, c'est-à-dire que l'image renvoyée par les points de la surface des objets est indépendante du point de vue. Ceci n'est en particulier pas vérifié lorsque les objets sont brillants.

L'hypothèse suivante, est une hypothèse de lissage. Celle-ci apparaît souvent dans les calculs. Elle sous-entend que le monde est constitué d'un ensemble de surfaces lisses et que deux points côte à côte ont donc plus de chance d'être à des profondeurs proches qu'éloignées. Enfin on doit également faire des hypothèses sur la qualité des images sur lesquelles on travaille et sur l'utilisation du matériel. Comme nous le verrons par la suite, il est tout d'abord important que les images

ne subissent pas d'effet de distorsion. Il faut ensuite que l'emplacement des caméras ainsi que leurs paramètres internes soient parfaitement connus, il s'agit des hypothèses de calibration. A celle-ci est liée celle de la géométrie équipolaire. Cela consiste à pouvoir faire correspondre les pixels d'une droite d'une image avec ceux d'une droite de l'autre image. Il faut pour cela que les plans formés par les centres optiques de chacune des caméras et les deux droites qui se correspondent soient confondus. Dans le cas le plus simple, les deux caméras sont à la même hauteur et sont légèrement décalées l'une de l'autre.



$$\mathbf{m}_1 = (x_1, y_1, 1), \mathbf{m}_2 = (x_2, y_2, 1), \text{ et } \mathbf{m}_1^T \mathbf{F}_{12} \mathbf{m}_2 = 0.$$

FIG. 2.1 – Schéma représentant la géométrie de l'expérience. On voit apparaître les deux plans images définis par les vecteurs u_1, v_1 et u_2, v_2 . Le plan équipolaire Π , et les deux droites mises en correspondance l_{m_1} et l_{m_2} .

2.2.1 Méthodes locales de base

Expliquons à présent le principe de la stéréoscopie. On rappelle que le but de la stéréoscopie est de retrouver la profondeur des pixels d'une image, soit encore d'évaluer le relief d'une scène. La stéréoscopie traditionnelle fait pour cela appel à

deux images prises à des points de vue très proches de façon à ce que les points qui apparaissent sur une image apparaissent également sur l'autre et réciproquement. Si on imagine par exemple un obstacle avec des objets derrière lui, si une des caméras est placée juste devant celui-ci et l'autre à côté. La première image ne verra que l'obstacle alors que la seconde fera apparaître ce qui se trouve derrière et les deux images ne pourront donc être mises en correspondance. Ce sont des problèmes d'occlusions. On parle d'occlusion lorsqu'il y a des parties cachées. On peut remarquer que lorsque les occlusions sont communes aux deux images, cela ne pose pas de problème.

Ayant ces deux images légèrement décalées l'une de l'autre, il nous faut alors trouver les correspondances entre les pixels de ces deux dernières. Pour cela on se sert de la géométrie équipolaire mise en place lors de l'expérience. Prenons l'image 1 comme image de référence et admettons pour simplifier le problème que les deux images soient prises à la même hauteur. On sait alors que pour un pixel de cette image, le pixel correspondant de l'autre image se trouve sur la droite située à la même hauteur. Et par ailleurs sensiblement à la même place. Dans le cas le plus simple et le plus direct, on comparera la couleur de tous les pixels de cette droite avec celle du pixel de l'image de référence. Celui qui aura la couleur la plus proche sera alors considéré comme le point correspondant.

Enfin, par une construction géométrique, l'intersection des deux droites formées par les deux pixels considérés et les centres optiques des deux caméras sera l'emplacement du point observé dans la scène réelle et on aura ainsi la profondeur du point dans le monde réel. On remarque que ces deux droites sont bien dans le même plan grâce à la géométrie équipolaire mise en place et donc qu'elle se coupe bien

en un point (celles-ci n'étant en théorie pas parallèles).

La démarche inverse et finalement la plus utilisée, consiste à prendre un pixel de l'image de référence et à le projeter dans le monde réel en choisissant arbitrairement une plage de valeurs pour la profondeur de ce point. On obtient donc une liste de points possibles et on reprojette ensuite ces points dans l'image 2. On obtient alors une liste de pixels dont on compare les couleurs avec celle du point de référence. La profondeur qui aura donné le couple de couleur le plus proche sera alors considérée comme la profondeur réelle. On procède ainsi avec l'ensemble des pixels de l'image de référence figure 2.2.

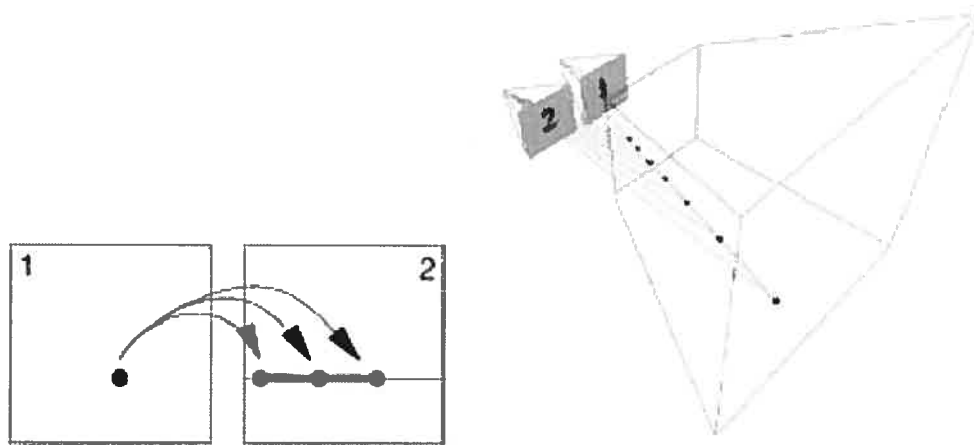


FIG. 2.2 – Illustration d'un point de l'image 1 projeté dans le monde réel à plusieurs profondeurs et reprojété sur l'image 2.

Il s'agit là d'une méthode locale. Retrouver la profondeur permettant de faire correspondre deux pixels des deux images revient à minimiser une fonction d'énergie qui ne dépendrait dans ce cas le plus simple que du coût de mise en correspondance. Ce coût étant proportionnel à la différence de couleurs entre deux pixels.

2.2.2 Méthodes globales

On distingue ensuite des méthodes globales qui sont moins directes et un peu plus avancées [RC98] [IG98] [IG93] [BI99]. On fait alors apparaître un terme de lissage dans la fonction d'énergie locale pour chaque pixel. Celle-ci se met alors sous la forme :

$$E(\text{pixel}) = E_{\text{correspondance}} + E_{\text{lissage}}(\text{pixel}, \text{pixels_voisins}) \quad (2.1)$$

Ce nouveau terme dépend de la différence de profondeurs entre celle du pixel étudié et celle de son voisin de droite et de gauche, dans le cas d'un système de voisinage de premier ordre. On se sert donc ici d'un *a priori* qui tend à faire disparaître les erreurs locales.

2.2.3 Méthodes dynamiques

On peut pousser plus loin cette méthode en allant vers la programmation dynamique [Vek99] [GLY92] [CHRM96] [OK85]. L'idée n'est plus de tenir compte seulement des voisins des pixels mais de prendre en compte l'ensemble des pixels d'une même droite. On cherche donc à minimiser l'énergie globale de cette droite. Si l'on étudie par exemple les pixels de gauche à droite, on peut pour chaque pixel, excepté le premier, prendre comme énergie :

$$E(\text{pixel}, \text{profondeur } d) = \min_{\text{profondeur } d'} [E_{\text{correspondance}} + E_{\text{lissage}}(\text{pixel}, \text{pixels_voisin}) \\ + E(\text{pixel_voisin}, \text{profondeur } d')] \quad (2.2)$$

Le pixel voisin est alors le pixel de gauche et pour chaque pixel, on enregistre une énergie par profondeur associée. Si on a une plage de n profondeurs possibles et une droite de m pixels, on a donc un tableau de $n \times m$ pixels. On voit donc que l'énergie dépend à la fois du lissage avec le plus proche voisin mais également de l'énergie optimale jusqu'alors calculée pour le pixel précédent. Une fois arrivé au dernier pixel le plus à droite, on prend la profondeur pour laquelle l'énergie est minimale et on remonte le tableau vers la gauche en prenant à chaque fois la profondeur retenue pour obtenir l'énergie minimale (soit la profondeur d' dans la formule).

On peut représenter ce problème sous la forme d'un graphe. Partons par exemple d'un tableau où les lignes correspondent aux différents pixels d'une ligne et les colonnes aux différentes profondeurs des pixels et chaque case correspond alors au coût de correspondance d'un pixel pour une profondeur. On remplace alors chaque case par un arc vertical ayant pour coût celui de la case, et on place un noeud entre chaque arc adjacent. On ajoute ensuite un arc entre deux noeuds situés à la même hauteur et côte à côte ayant pour valeur un coût de lissage entre deux profondeurs intermédiaires. On peut alors utiliser le théorème du flot maximal dans un graphe afin de déterminer la coupure minimale qui correspondrait à l'ensemble des profondeurs pour chaque pixel, qui minimise l'énergie de la droite (voir exemple figure 2.3).

On a donc vu ici les principes de base de la stéréovision ainsi que les principaux types de méthodes : locale, globale, dynamique. Il existe au sein de celle-ci de nombreuses variantes et adaptations particulières. La façon de calculer les coûts d'énergie n'est par exemple pas unique. Nous sommes également partis d'un problème à deux images, mais il est très courant d'en utiliser plus afin de minimiser les erreurs. Bien sûr le coût de calcul se trouve alors plus élevé et un excès d'images

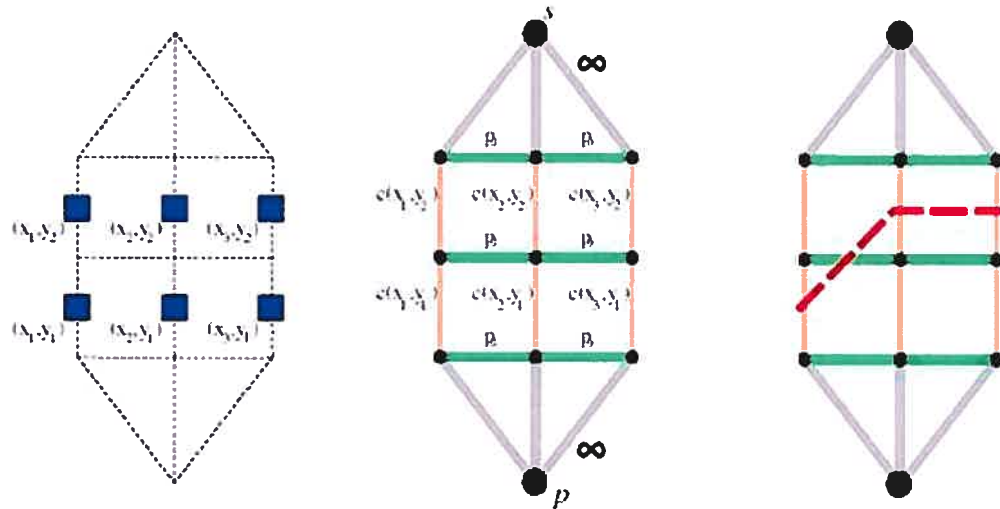


FIG. 2.3 – Exemple d'un graphe pour minimiser l'énergie d'une droite de trois pixels avec deux profondeurs possibles à chaque fois. A gauche, on fait apparaître les cases du tableau correspondant au coût. Au milieu le graphe associé où les arcs horizontaux correspondent à des coûts de lissage. A droite un exemple de coupure minimale. Les profondeurs retenues pour chaque pixel correspondent aux arcs verticaux coupés.

peut inversement créer des erreurs.

Rejoignant l'idée de la reconstruction en trois dimensions à partir d'images et de mises en correspondance nous allons à présent voir des méthodes dites de « coloriage de voxels ».

2.3 Méthode de coloriage de voxels

Dans le cas de la stéréoscopie classique, nous avons vu que les images devaient être très proches afin que tout ce qui est présent sur une image soit présent également sur l'autre image et réciproquement. Le désavantage de ceci est qu'alors, deux

points situés sur le même axe dans le monde réel (même x et même y dans le repère de la caméra) mais avec des profondeurs différentes se projettent pratiquement au même endroit entraînant des erreurs dans le calcul des profondeurs (voir schéma 2.4).

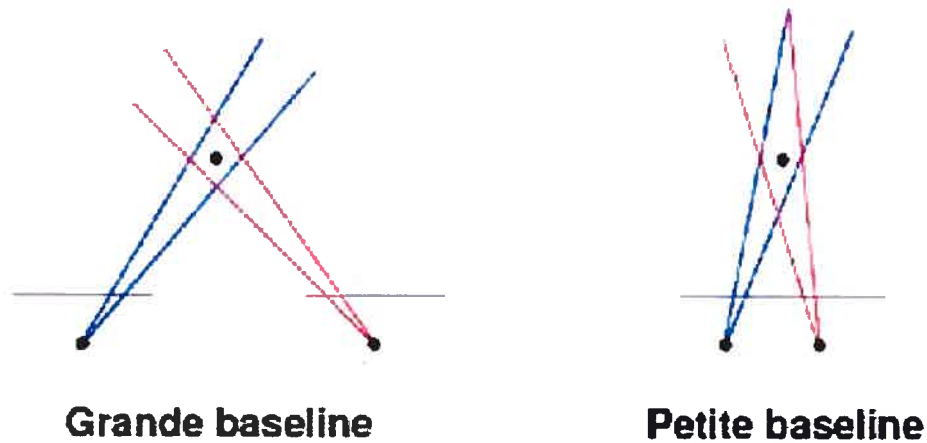


FIG. 2.4 – Schéma illustrant le cas d'une petite et d'une grande "baseline".

Il s'agirait donc de pouvoir prendre des images à des points de vue éloignés tout en palliant au problème des images qui seraient alors très différentes bien qu'axées sur la même scène. C'est vers cette idée que vont les méthodes de coloriage de voxels ou encore de stéréo-volumétrique.

Commençons par quelques définitions :

On appelle un voxel photo-cohérent, un voxel (c'est-à-dire un point de l'espace) qui est vu de la même façon, avec la même radiance, par l'ensemble des caméras qui le voient. Une scène est alors dite photo-cohérente si l'ensemble des voxels de sa surface sont photo-cohérents. Dans notre présentation, on considèrera un volume

de départ de la forme d'un cube constitué de voxels et dont la longueur de l'arête fait n voxels. Ce volume contient la scène ou l'objet étudié. Si il y a C couleurs possibles pour un voxel, il y a donc C^{n^3} scènes possibles.

Dans le cas le plus simple, on recherche juste la silhouette de l'objet étudié. On a alors $C = 2$, on ne considère que le vide ou l'objet. Chaque caméra permet alors de définir une sorte de cône dans le lequel l'objet est contenu. Il reste alors à prendre l'intersection de tous les cônes pour obtenir la reconstruction de l'objet étudié. Bien sûr la précision obtenue est très faible et n'est vraiment valable que pour des formes convexes dans le cas en deux dimensions. On n'obtient qu'une sorte d'enveloppe et la qualité de celle-ci dépend directement du nombre de caméras disposées (voir figure 2.5). Bien que très peu coûteuse cette méthode est très peu utilisée pour la reconstruction, cependant on peut s'en servir comme étape préliminaire avant d'utiliser un algorithme plus complexe et plus fin.

L'étape suivante est de ne plus avoir de contrainte sur la couleur, on ne se limite plus au vide ou au non-vide mais on impose une contrainte sur les caméras. Il faut que celles-ci soient placées de telle sorte que l'ordre de profondeur des voxels du volume soit indépendant de la caméra. Le principe est ensuite simple, on prend les voxels un par un en commençant par les plus proches et en s'éloignant petit à petit. Pour chaque voxel, on le projette sur toutes les caméras et on calcule la cohérence obtenue. Si ce pixel est photo-cohérent, on le garde, sinon on le retire du volume. On procède ainsi pour tous les voxels du volume et on obtient au final la reconstruction en trois dimensions de la forme recherchée [SD97] [THS04] (voir figure 2.6).

Ce procédé reste assez simple et produit de bons résultats. On peut cependant

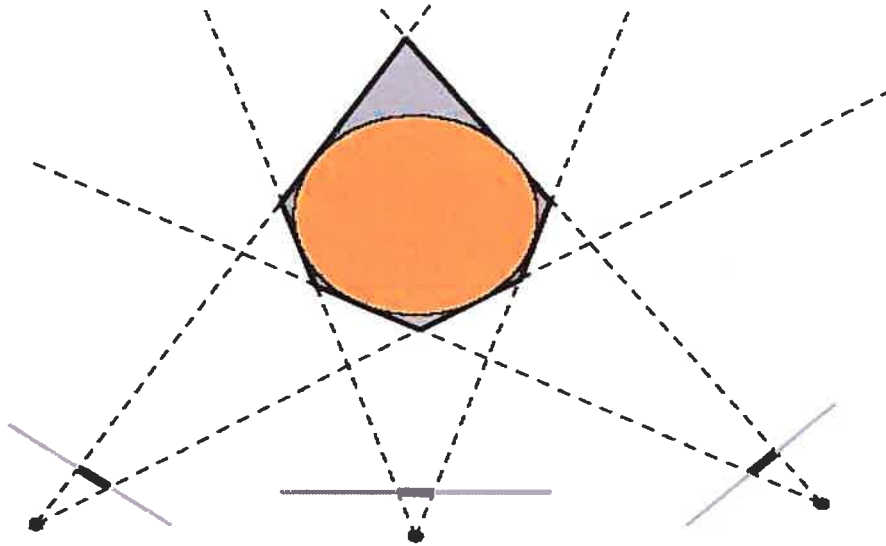


FIG. 2.5 – Coloriage de voxels dans le cas le plus simple. La forme étudiée est le disque orange et la forme obtenue est la partie grisée. Celle-ci est approximative et n'est acceptable que lorsque la forme est convexe et en deux dimensions.

généraliser la méthode en plaçant les caméras sans contraintes. C'est la méthode appelée « space carving » ou encore « creusage de l'espace » et a été conçue par Kutulakos et Seitz [KS00]. Le principal changement est que le calcul est plus lourd. Pour chaque voxel, il est plus compliqué de savoir si celui-ci est visible ou non par une caméra, et lorsque l'un d'eux est supprimé, car non photo-cohérent, la mise à jour que cette suppression entraîne est plus lourde. Une solution pratique consiste à faire un balayage méthodique plan par plan en procédant dans les six directions principales et en ne considérant à chaque fois que les caméras situées du côté extérieur par rapport au plan étudié (voir figure 2.8). Citons également d'autres articles traitant de cette méthode : [Kut00] [SCMS04] [SCMS99] [SSH02]. Enfin, deux articles effectuent un travail de comparaison de différentes méthodes aussi bien en stéréovision classique qu'en méthode de coloriage de voxels : [SSZ01] [SCD⁺06].

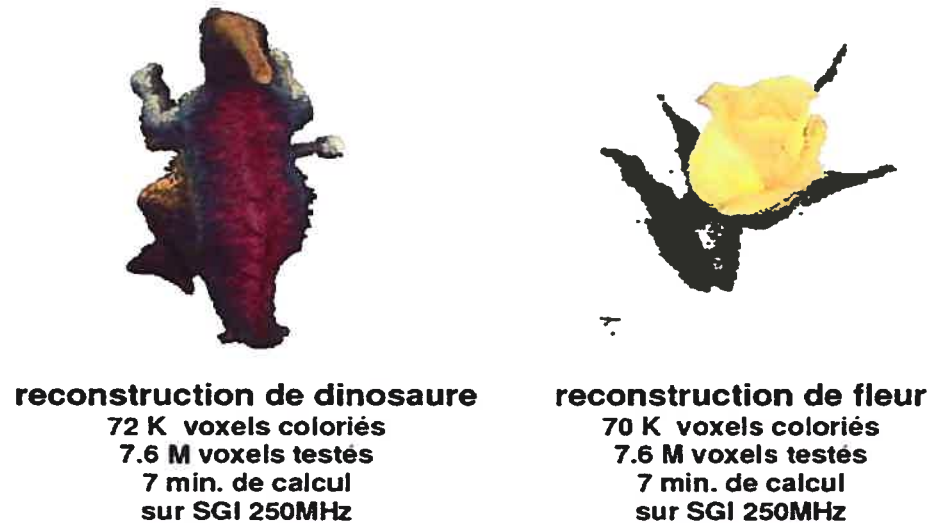


FIG. 2.6 – Exemple de résultats obtenus par coloriage de voxels avec contrainte sur les caméras.



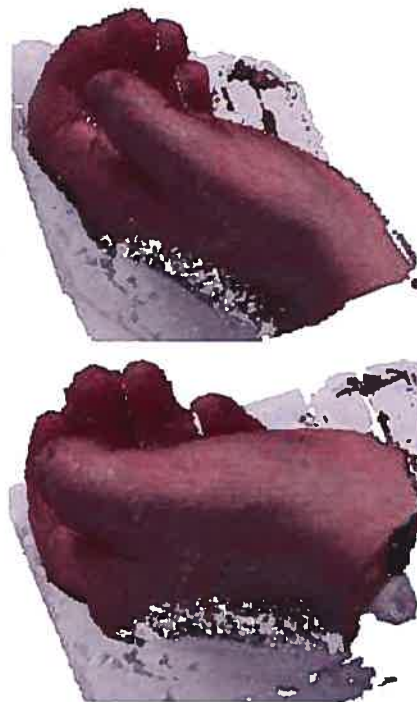
FIG. 2.7 – Schéma d'un volume que l'on "creuse" par la méthode du "space carving".

2.4 Méthode de reconstruction basée sur de forts *a priori*

Les méthodes de reconstruction que nous avons vues jusqu'à présent sont des méthodes tout à fait générales et même si elles se basent sur certains *a priori*,



**Image d'entrée
(1 sur 100)**



Vues de la reconstruction

FIG. 2.8 – Exemple de résultat obtenu par "space carving"

ceux-ci sont globalement vérifiés ou il peut être fait en sorte qu'ils le soient. Cependant, dans certains cas bien précis, lorsque l'on a de fortes connaissances sur l'objet que l'on cherche à reconstruire, il est alors très intéressant de se servir de ces données. On peut alors à la fois réduire les calculs et améliorer la reconstruction.

Prenons l'exemple d'un procédé de reconstruction utilisé dans le domaine médical. Il s'agit de redessiner en trois dimensions une colonne vertébrale afin de traiter des cas de scoliose. On procède pour cela en deux étapes en se servant d'un certain nombre de pré requis réduisant ainsi très fortement le nombre de détails à connaître [BMLG05].

La première étape consiste à faire une première reconstruction grossière de la colonne. Les données sur lesquelles on travaille sont deux radiographies, une frontale et une latérale. Pour simplifier le problème et rendre la solution meilleure, on demande à l'utilisateur de pointer l'extrémité inférieure et supérieure de la première vertèbre sur les deux vues. A partir de là, on est capable de déterminer l'échelle et l'orientation des données et on modifie alors l'ensemble des modèles que l'on a à notre disposition de façon à ce que ces derniers aient la même taille et le même angle pour permettre les comparaisons.

Une fois cette première approximation faite, on cherche alors à rendre le modèle plus précis en travaillant vertèbre par vertèbre. On collecte pour cela une base de données de l'ensemble des vertèbres de la colonne sur un échantillon important d'individus étant atteints ou non de scoliose. On identifie ensuite pour chaque vertèbre, un nombre de paramètres permettant de les caractériser. Dans un espace de dimension égal au nombre de paramètres, on peut alors faire divers regroupements de points et ainsi définir des formes « type ». L'ensemble de ces formes constitue alors une base de données qui permet de reconstruire n'importe quelle colonne vertébrale étudiée. Les données qu'on a alors besoin d'obtenir pour effectuer les reconstructions sont alors très limitées. Il suffit à chaque fois d'identifier certains paramètres simples qui peuvent être reconnus avec peu de coupes radiographiques. On fait ainsi bien apparaître que ce procédé permet grâce à des pré requis, de reconstruire des formes très complexes à partir d'un modèle paramétrique concis (voir exemple figure 2.9).

Dans le même ordre d'idée, une autre étude a été faite pour reconstruire les cages thoraciques [BMDG05]. Cette fois-ci, l'*a priori* de départ est encore plus

fort. Le travail principal se fait sur le choix et le regroupement des paramètres qui vont permettre une classification des différents types de cages thoraciques. A partir d'un certain nombre de critères, grâce à une analyse en composantes principales faite sur des données préalablement recueillies sur un échantillon de population de cages thoraciques, on va chercher à faire des regroupements et définir des classes de formes. Chaque groupe de formes est donc caractérisé par une plage de valeurs propres associées à des déformations statistiquement admissibles. Lors de la reconstruction en trois dimensions d'une cage thoracique, il s'agit de déterminer à quel type de cage on a à faire. On détermine une fonction d'énergie d'adéquation qui utilisera les contours projetés latéralement et frontalement qui fera apparaître les différences entre la cage étudiée et celles des différentes classes. On retiendra celle pour laquelle la fonction d'énergie est la plus petite (voir exemple figure 2.10). Encore une fois ce type de reconstruction est très éloigné des méthodes générales mais présente un autre aspect intéressant en montrant que dans des cas précis où l'on peut faire appel à suffisamment de pré requis (base d'apprentissage) il est beaucoup plus avantageux d'utiliser ce dernier type de méthode.

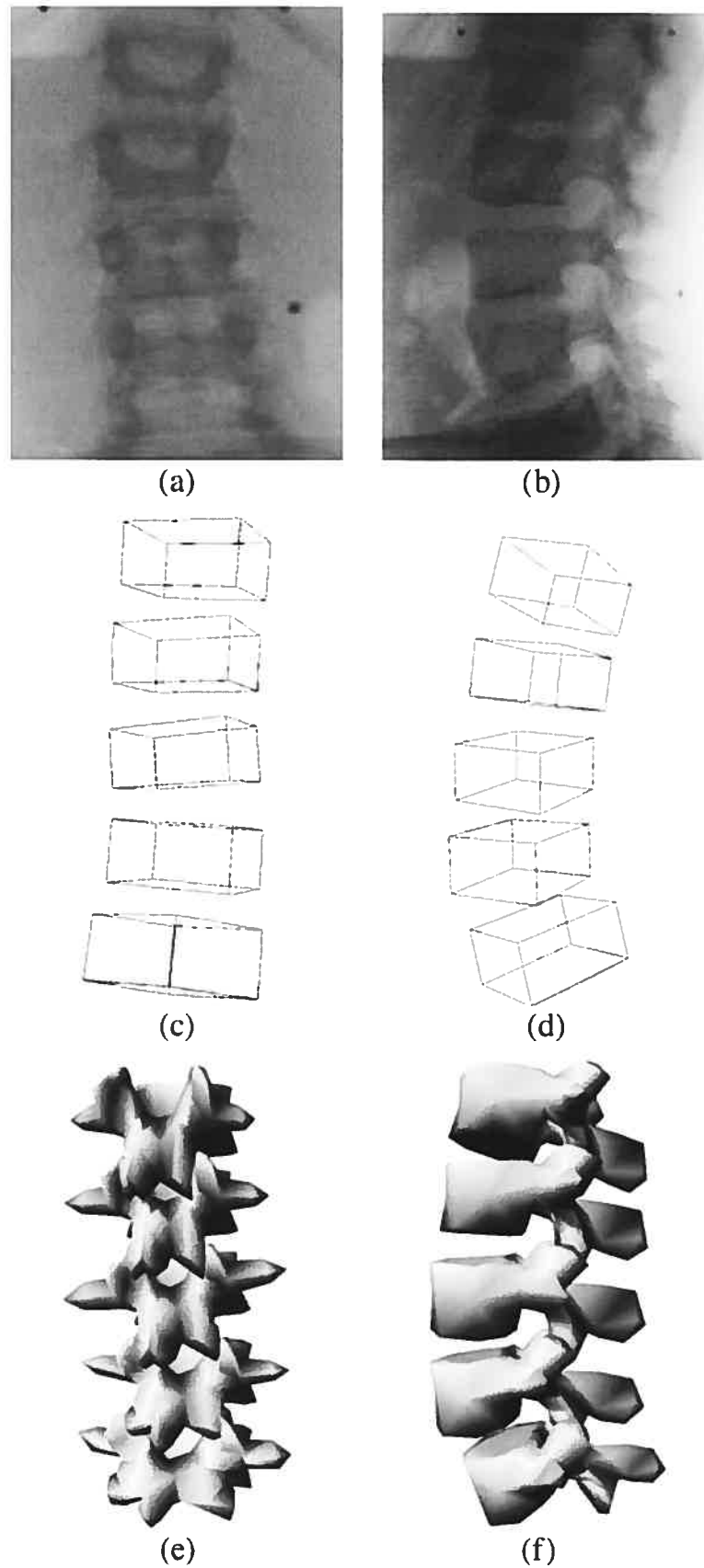


FIG. 2.9 – Exemple de reconstruction obtenue à partir de deux radiographies.

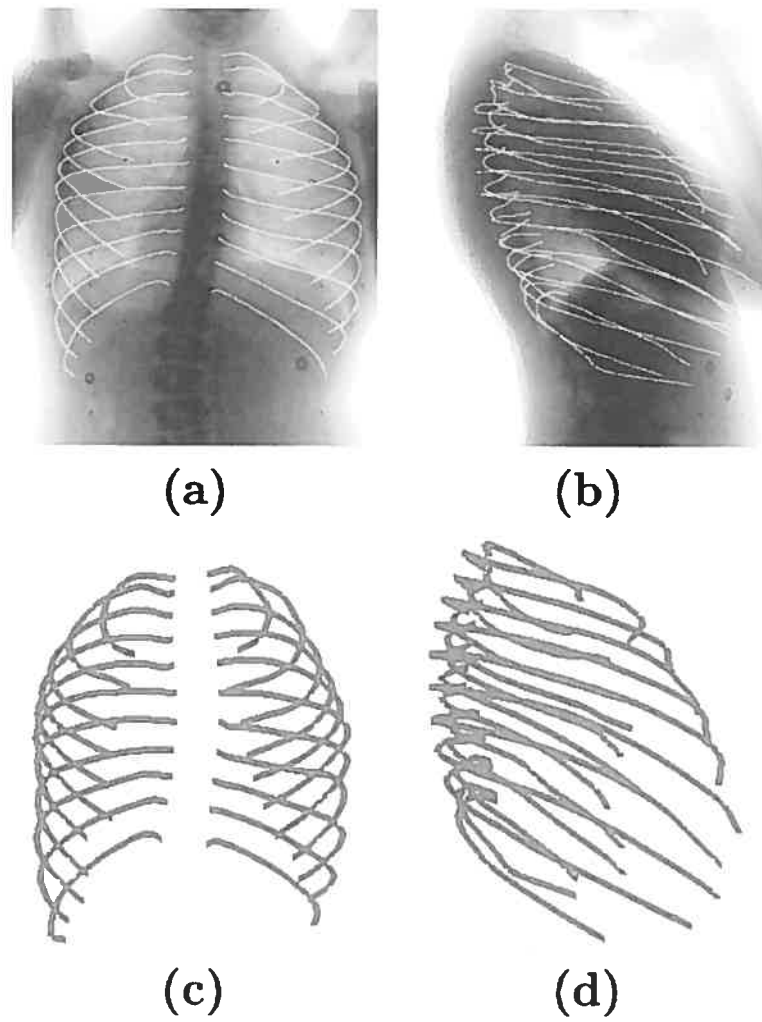


FIG. 2.10 – Exemple de reconstruction de cage thoracique. (a) et (b) sont les radiographies sur lesquelles on travaille. (c) et (d) sont les résultats retenus.

CHAPITRE 3

RECONSTRUCTION POLYÉDRIQUE À PARTIR DE CARTES DE PROFONDEURS

Nous avons donc vu, dans un premier temps, les méthodes générales de reconstruction en trois dimensions qui sont les plus utilisées dans ce domaine et qui se basent sur la reconstruction d'un volume en étudiant l'ensemble de ses voxels. Nous avons ensuite vu quelques exemples d'applications plus spécialisées et adaptées à des problèmes particuliers. Revenons à présent sur des algorithmes généraux mais qui se basent non plus sur une recherche par voxel, mais sur une reconstruction du contour en partant d'une forme initiale polyédrique et utilisant des séries de coupes projectives [SMD04].

3.1 Principe

Le principe est le suivant : il s'agit tout d'abord de recueillir des cartes de profondeurs d'un objet sous plusieurs directions. Celles-ci sont dans notre cas obtenues pour chaque direction par un couple de cartes de profondeurs de la vue avant et arrière de la forme. Ces données sont discrètes et pour chaque direction, on a donc une « grille » de valeurs. On part ensuite d'une forme initiale sur laquelle on va calculer ses cartes de profondeurs sous les mêmes directions que celles obtenues sur l'objet que l'on veut reconstruire. On cherche alors à minimiser la fonction de différence de ces deux jeux de données. A cette fin, on utilise un algorithme itératif de descente par gradient. A chaque itération on déplace les points de la forme polyédrique que l'on travaille, réduisant ainsi les écarts sur les données. Une fois que

la différence est en dessous d'un certain seuil, on en déduit que l'on a retrouvé la forme de l'objet étudié.

3.2 Résumé de l'algorithme principal

Comme nous venons de le mentionner, le problème de reconstruction est réduit à un problème de minimisation de fonction d'énergie à l'aide d'une descente de gradient. L'énergie à minimiser est la différence des projections de la forme étudiée avec celles de la forme que l'on modèle. Si on note $d_{\theta,i,j}$ la valeur de la projection aux coordonnées i, j pour l'angle de vue θ de la forme étudiée, $p_{\theta,i,j}$ celle de la forme que l'on modèle et v l'ensemble des points de celle-ci, on a alors comme fonction d'énergie :

$$E(v) = d \sum_{\theta,i,j} (p_{\theta,i,j}(v) - d_{\theta,i,j})^2 \quad (3.1)$$

On remarque que d ne dépend pas de v car il s'agit des données que l'on obtient au départ sur la forme que l'on veut retrouver et qui reste donc inchangées. Le gradient de cette fonction est alors :

$$E(v) = \begin{matrix} \frac{\partial E}{\partial v_0}(v) \\ \dots \\ \frac{\partial E}{\partial v_n}(v) \end{matrix}$$

avec :

$$\frac{\partial E}{\partial v_i}(v) = \sum_{\theta,i,j} 2 \frac{\partial p_{\theta,i,j}}{\partial v_i}(v) (p_{\theta,i,j}(v) - d_{\theta,i,j})$$

$d_{\theta,i,j}$ est connu au départ. Les deux termes inconnus sont donc $p_{\theta,i,j}(v)$ et $\frac{\partial p_{\theta,i,j}}{\partial v_i}(v)$. $p_{\theta,i,j}(v)$ n'est pas très complexe à calculer, contrairement à $\frac{\partial p_{\theta,i,j}}{\partial v_i}(v)$.

Il s'agit alors de procéder par étapes pour parvenir à déterminer cette dérivée partielle. La première idée est la suivante : il faut tout d'abord remarquer que lorsque l'on bouge un point de notre forme, le seul volume modifié est celui formé par le point en question et ses voisins directs (voir figure 3.1). Dès lors, durant le calcul de la dérivée on ne s'intéresse alors qu'à ce seul volume. Ensuite, pour une droite de projection, on observe si celle-ci coupe ce volume ou non. Si elle ne le coupe pas, la dérivée est nulle. Sinon on décompose le volume en plusieurs petits volumes élémentaires constitués de quatre points. Nous verrons que pour cette décomposition, nous utiliserons un point supplémentaire qui sera en général le barycentre des voisins du point par rapport auquel on dérive. On étudie un à un ces volumes, sachant qu'un calcul est nécessaire à chaque fois que la droite de projection coupe le volume considéré. Lorsque cela est le cas, on exprime la valeur de la projection en fonction des coordonnées des points et on dérive cette expression par rapport aux coordonnées du point par rapport auquel on dérive. Ce calcul se divise également en plusieurs étapes et en plusieurs cas qui seront développés par la suite.

3.3 Principe de la tomographie

Avant d'entrer au coeur de notre méthode, commençons tout d'abord par donner quelques détails concernant le type de données sur lesquelles nous allons travailler. Il s'agit de données tomographiques. Le principe de la tomographie est assez simple. Il consiste de façon très générale à recueillir des groupes de données sur un objet dans le but final de reconstruire cet objet. Ces données se caractérisent du fait qu'elles s'obtiennent le long de droites toutes parallèles pour un même groupe. Si

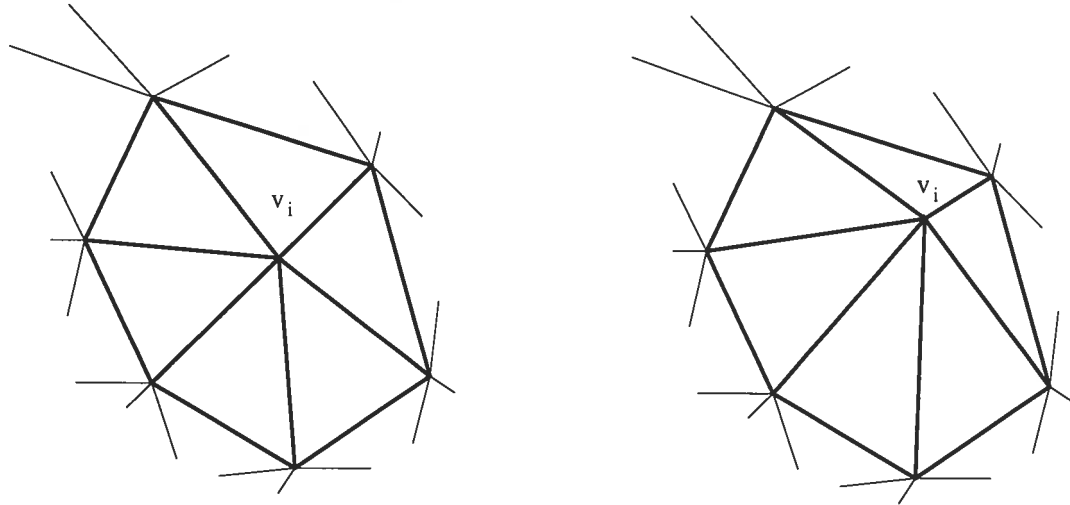


FIG. 3.1 – On fait ici apparaître un point v_i de la forme avec ses voisins (figure de gauche). On déplace ensuite celui-ci et on observe que seul le volume défini par le point v_i et ses voisins (représenté en gras) est modifié. Le reste de la forme est inchangée. Le calcul de dérivation ne se fera donc qu'en étudiant ce volume.

l'on peut imaginer plusieurs type de données, celles-ci sont en général liées à la quantité de matière traversée par chacune de ces droites. Ainsi si on imagine un objet au milieu d'un repère (O, x, y, z) et un ensemble de droites toutes parallèles à l'axe (O, z) et quadrillant le plan (O, x, y) , on obtient donc une grille de valeurs où chaque valeur correspond à l'épaisseur traversée par la droite correspondante à travers l'objet. Ce qu'il est important de faire ressortir dans cette méthode, c'est que l'on ne reconstruit pas directement l'objet comme on a pu le voir dans le cas de la stéréovision ou de du "space carving". De plus, un groupe de données dans une seule direction ne suffit jamais à reconstruire une forme. Il faut toujours utiliser au moins deux groupes de données dans des directions différentes et ensuite travailler dessus, c'est-à-dire les mettre en correspondance de façon à effectuer la reconstruction.

Observons un exemple simple illustrant le manque d'information lorsque l'on a

des données dans seulement une direction. En observant l'exemple en deux dimensions, figure 3.2, on observe que deux formes différentes auront rigoureusement les mêmes données tomographiques dans certaines directions.

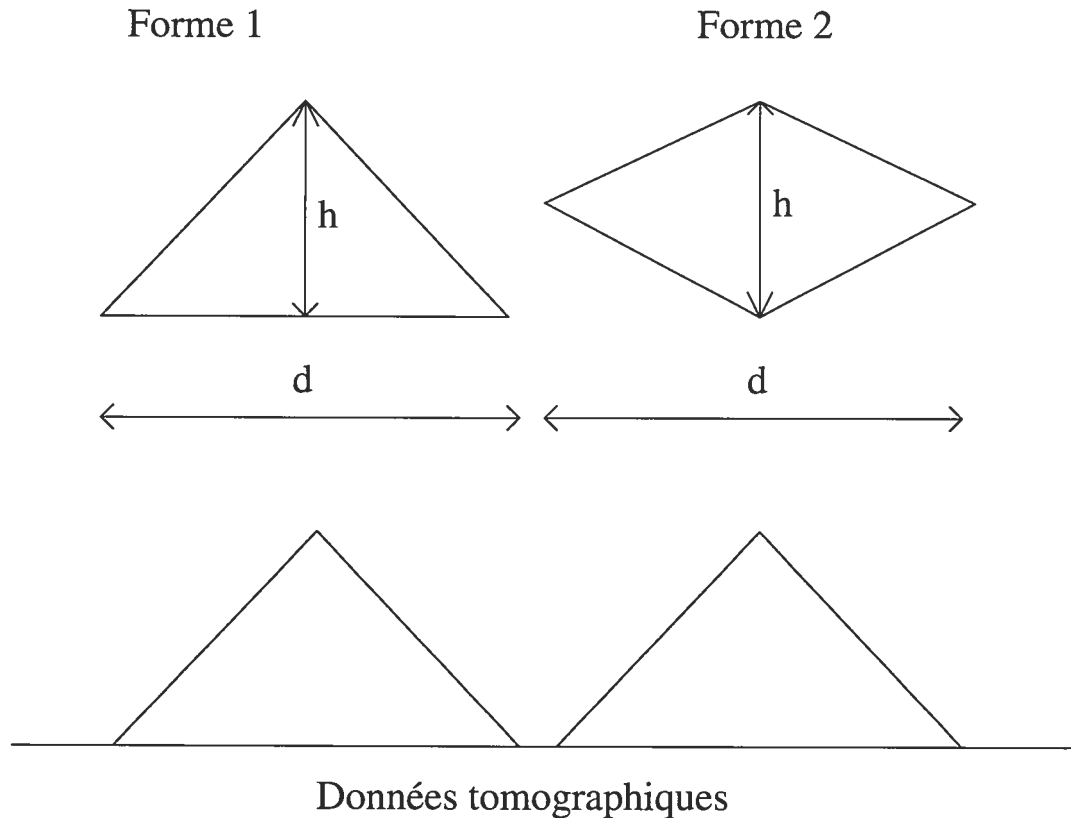


FIG. 3.2 – Un triangle isocèle de base d et de hauteur h et un losange de diagonale d et h . Projeté sur une direction verticale, ces deux formes engendrent les mêmes données tomographiques.

En revanche, des données qui seraient prises dans une direction orientée à 90 degrés de la première, donneraient des données différentes et rassembler les deux jeux de données permettrait de reconstruire chacune de ces formes.

Pour des formes plus complexes, on peut avoir besoin de plus de vues encore. Comme nous venons de l'expliquer, les données tomographiques ne sont pas di-

rectement utilisables, il faut les traiter pour effectuer une reconstruction. Il existe pour cela plusieurs méthodes. On peut par exemple passer ces données dans le domaine fréquentiel et travailler dans celui-ci. Dans le cas de notre méthode nous nous distinguons du fait que l'on ne traite pas vraiment ces données mais on les compare avec celles de la forme que l'on modèle dans le but de retrouver la forme cible, ce qui rend la méthode d'autant plus originale et intéressante.

Voyons à présent en détail les différentes étapes de notre méthode de reconstruction.

3.4 Obtention des cartes de profondeurs

Commençons tout d'abord par nous intéresser à l'acquisition des cartes de profondeurs de type tomographique. La méthode sur laquelle on se base [SMD04] récupère ces données par rayons X. Une palette d'émetteurs envoie donc des rayons qui vont traverser l'objet étudié et des capteurs situés de l'autre côté « reçoivent » ces rayons et en fonction de l'énergie reçue, on en déduit la quantité de matière traversées par ces derniers. Plus un rayon traverse de la matière, plus son énergie se voit amortie. L'utilisation de rayons X apporte plusieurs avantages. D'une part, cela va permettre de reconstruire des formes qui ne sont pas visibles mais présentes à l'intérieur d'un corps. On peut par exemple repérer une bulle d'air à l'intérieur d'un métal. À condition de connaître la densité du métal en question et celui de l'air ou du gaz quel qu'il soit, le calcul est le même qu'un objet situé dans du vide. Un autre avantage important est que, contrairement à une étude visuelle, les intégrations le long des rayons fonctionnent également avec des formes non convexes comme l'illustre le schéma de la figure 3.3.

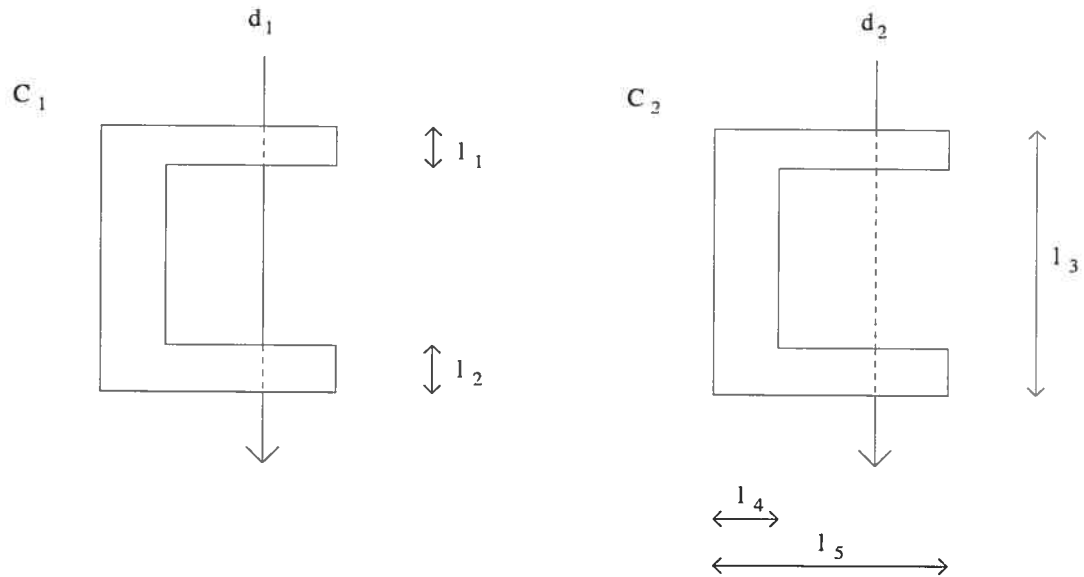


FIG. 3.3 – Sur la forme C_1 , la projection le long de d_1 est celle obtenue par un rayon, la valeur rapportée sera donc $l_1 + l_2$. Dans le cas d'une mesure visuelle, forme C_2 , seuls les premiers obstacles au-dessus et en-dessous sont détectés, on ignore ce qu'il y a entre. La valeur enregistrée est alors l_3 , celle-ci est fautive et sera de plus en contradiction avec des projections faites dans d'autres directions. Par exemple une projection horizontale enregistrera une valeur l_4 alors que la mesure verticale l_3 sous-entend que l'on devrait trouver la valeur l_5 (ce que l'on trouverait si la pièce était pleine).

En revanche, ce procédé a également ses défauts. Il est d'une part très coûteux car il s'agit d'une technologie de pointe. De plus, la précision des rayons n'est pas parfaite, ils ne sont pas ponctuels et peuvent subir des déviations partielles. Un capteur ne reçoit donc pas uniquement les données issues de l'émetteur situé en face de lui. De plus, l'acquisition de telles cartes est longue : plusieurs heures, et dans certains domaines comme le domaine médical, on cherche à limiter autant que possible les radiations émises sur les patients et donc à limiter le nombre de radiographies à effectuer.

Dans notre étude, nous allons nous intéresser à des données obtenues visuelle-

ment. Il s'agit pour cela de travailler pour chaque direction, sur une vue avant et une vue arrière et de regrouper ces deux informations. On pourra obtenir ces cartes de profondeurs ou par stéréoscopie, ou, pour l'étude théorique, synthétiquement à partir de formes modélisées par ordinateur et à partir de logiciels ou de bibliothèques graphiques comme OpenGL, obtenir la profondeur des pixels représentés à l'écran. Cette solution visuelle pose des problèmes pour les formes non-convexes comme nous l'avons vu précédemment à la figure 3.3, en revanche, les dispositions pour ce type de mesures sont beaucoup moins coûteuses et le calcul très rapide, au pire des cas de l'ordre de quelques minutes. C'est pourquoi en fonction du contexte et des attentes recherchées, cet axe d'étude peut se montrer très intéressant.

L'intérêt de cette méthode n'étant pas dans la façon d'obtenir ces cartes mais réside plus dans l'étude des résultats que l'on peut obtenir avec ce type de cartes, se différenciant de celles obtenues par rayons X. On ne détaillera donc pas ici l'implémentation d'un code de stéréoscopie ou celle d'un code OpenGL.

3.5 Algorithme principal

3.5.1 Définition des lignes de "projection"

Considérons à présent le jeu de données acquis. Pour une direction donnée, on a donc une grille de valeurs. Soit $n \times n$ ce nombre de valeurs, et $p_{i,j}$ la valeur de l'intégration le long de la droite passant par le point de coordonnées $i, j, 0$ et parallèle à l'axe (O, z) . $p_{i,j}$ vaut donc 0 lorsque la droite concernée ne traverse en aucun point l'objet étudié. On notera une telle droite L_{ij} .

On a alors :

$$p_{i,j} = \int_{L_{i,j}} f(x, y, z) dl \quad (3.2)$$

Où f vaut 0 en dehors de l'objet et 1 à l'intérieur de celui-ci.

3.5.2 Initialisation de la forme de départ

Il nous faut ensuite partir d'une forme initiale. Cette forme est constituée d'un ensemble de points formant entre eux des facettes triangulaires. Celle-ci doit être la plus neutre et la plus régulière possible afin d'avoir la forme la plus générale. Il apparaît donc logique de partir d'une forme sphérique. Comme nous le verrons par la suite, celle-ci doit être définie par des points et les faces doivent être triangulaires. La "sphère" la plus simple possible est alors constituée de quatre points formant quatre triangles équilatéraux. Pour simplifier le calcul des coordonnées de ces points, on remarquera que la sphère s'inscrit dans un cube comme illustré dans la figure 4.5.

Pour augmenter le nombre de points, et donc travailler avec une forme plus précise, on procédera par itérations de la façon suivante : pour chaque face triangulaire, on divise celle-ci en quatre triangles équilatéraux en ajoutant des points au milieu des arêtes comme illustré figure 3.7.

Les nouveaux points ajoutés sont alors projetés sur la surface de la sphère que l'on cherche à représenter. A titre d'indication, on a donc pour 0 itération : 4 faces, 4 points, pour 1 itération : 16 faces, 10 points, pour 2 itérations : 64 faces, 34 points, pour 3 itérations : 256 faces, 130 points et pour 4 itérations : 1024 faces, 514 points. Lors de la création de cette forme, on veillera à enregistrer à la fois la

liste des points, avec un numéro pour chacun, leurs coordonnées et celles de leurs voisins, ainsi que la liste des faces avec les numéros de leurs points dans un sens trigonométrique et leur vecteur normal orienté vers l'extérieur.

Pour clarifier les notations, on notera $d_{i,j}$ les données obtenues sur l'objet cible et qui resteront inchangées tout au cours de l'algorithme, et $p_{i,j}$ celles de la forme sur laquelle on travaille. Plus généralement on pourra utiliser les notations $d_{\theta,i,j}$ et $p_{\theta,i,j}$ pour préciser la direction selon laquelle ont été faites les projections.

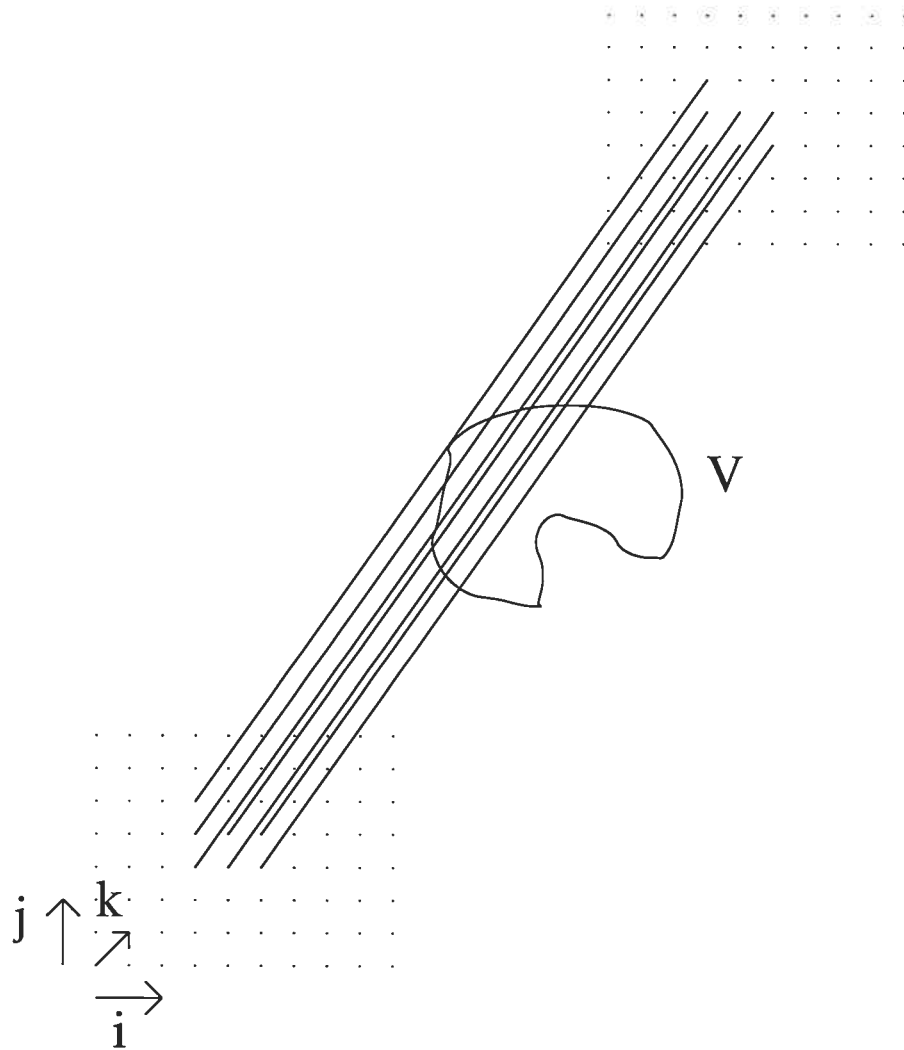


FIG. 3.4 – Illustration des lignes de projection d'équation $x = i$ et $y = j$ à travers le volume V . Par soucis de visibilité, on n'a représenté qu'une partie de ces lignes.

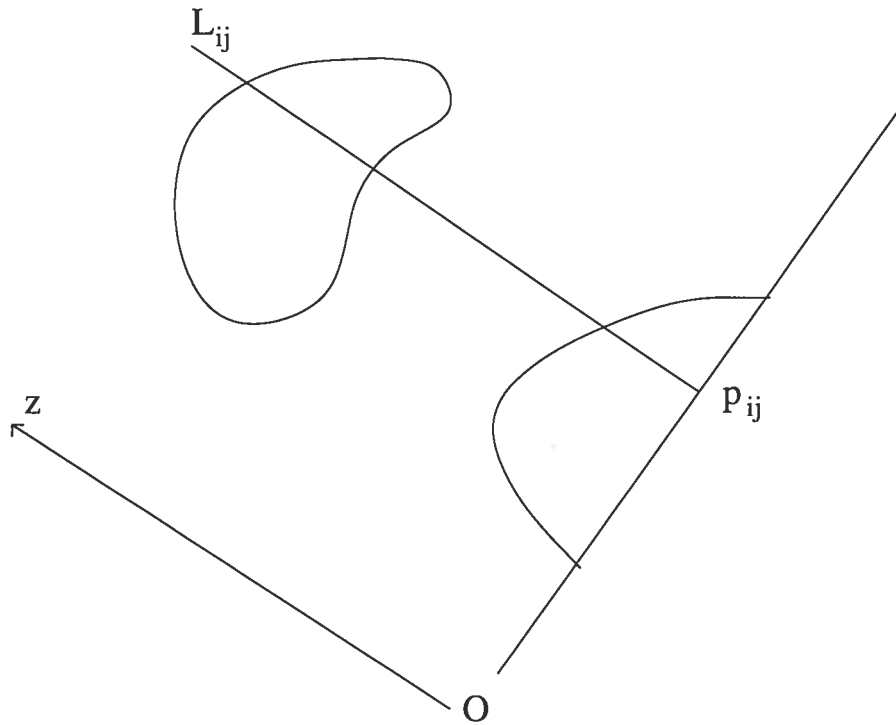


FIG. 3.5 – Schéma illustrant les données calculées, on pourra par abus de langage parler de données de projection même si il s'agit d'intégration le long de droite dans l'espace.

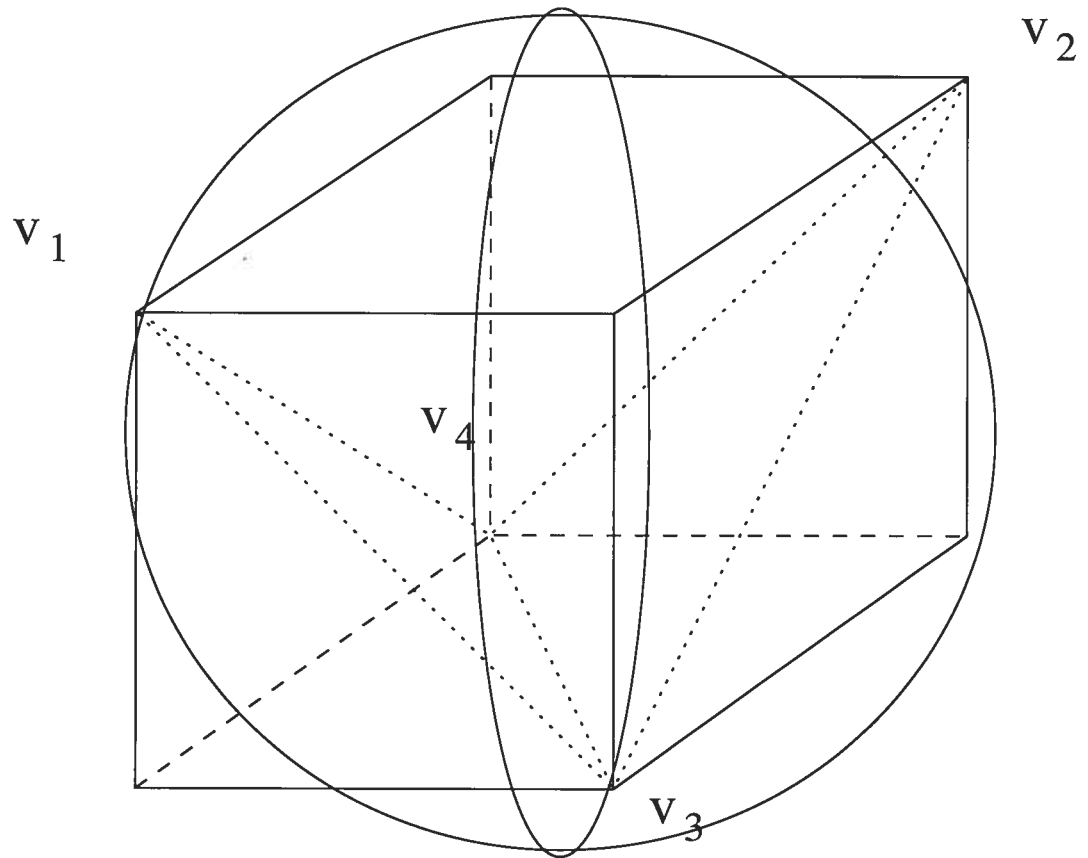


FIG. 3.6 – On voit ici la « sphère » représentée par quatre points : v_1 , v_2 , v_3 et v_4 inscrite dans un cube.

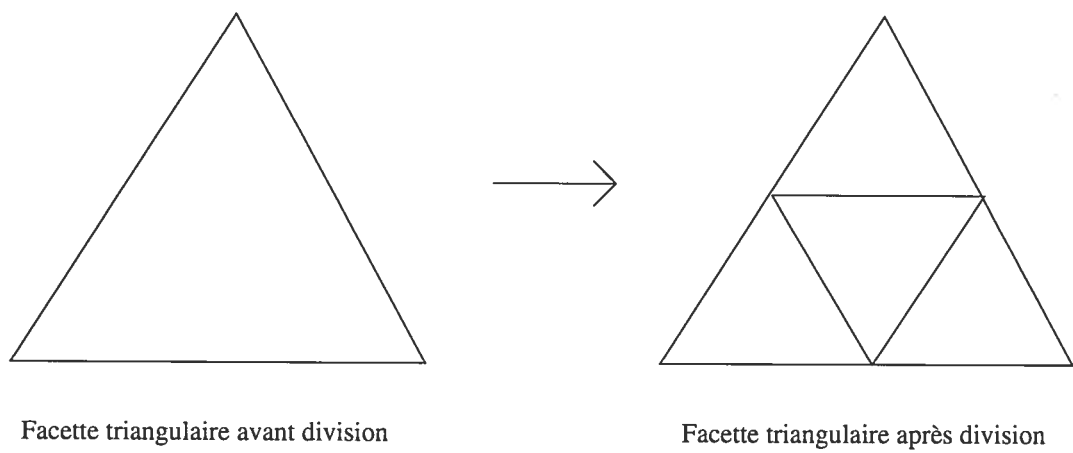


FIG. 3.7 – Illustration de la division d'une facette triangulaire en quatre autres facettes triangulaires.

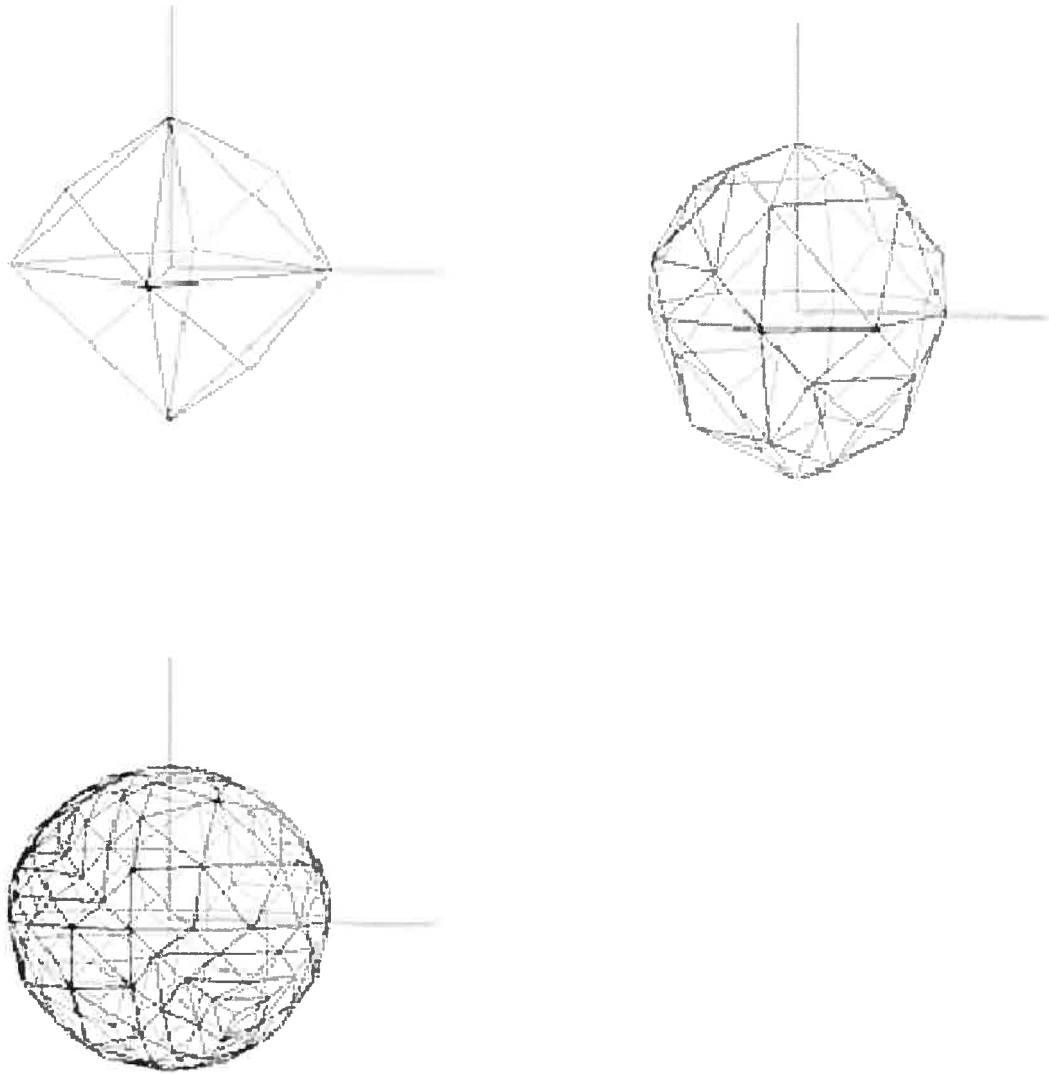


FIG. 3.8 – Sphères obtenues après une, deux et trois itérations.

3.5.3 Expression du gradient de la fonction d'énergie

On a donc à présent l'ensemble des $d_{\theta,i,j}$ et des $p_{\theta,i,j}$. On cherche donc à minimiser :

$$E(v) = d \sum_{\theta,i,j} p(v)^2 = \sum_{\theta,i,j} d_{\theta,i,j} p_{\theta,i,j}(v)^2 \quad (3.3)$$

Où v est un vecteur contenant l'ensemble des points de la figure.

Pour des raisons de temps de calcul et de simplicité d'implémentation, on va utiliser un algorithme de descente de gradient pour minimiser la fonction d'énergie. En notant v^k l'ensemble des points à la k ième itération et v^{k+1} celui à la $k + 1$ ième itération et g^k le gradient de E par rapport aux points de v^k , on a :

$$v^{k+1} = v^k - c g^k \quad (3.4)$$

Où c est un coefficient qui sera fixé arbitrairement. Dans la pratique nous lui donnerons une valeur comprise entre 0.01 et 0.001. Plus celui-ci est petit plus la précision est grande mais plus le nombre d'itérations doit être important pour faire diminuer la fonction d'énergie. Et

$$g^k = \nabla E(v^k) = \begin{pmatrix} \frac{\partial E}{\partial v_0^k}(v^k) \\ \dots \\ \frac{\partial E}{\partial v_n^k}(v^k) \end{pmatrix} \quad (3.5)$$

et

$$\frac{\partial E}{\partial v_i^k}(v^k) = \frac{\partial}{\partial v_i^k} \sum_{\theta,i,j} d_{\theta,i,j} p_{\theta,i,j}(v^k)^2 = \sum_{\theta,i,j} \frac{\partial d_{\theta,i,j}}{\partial v_i^k} p_{\theta,i,j}(v^k)^2 \quad (3.6)$$

Soit encore :

$$\frac{\partial E}{\partial v_i^k}(v^k) = 2 \sum_{\theta, i, j} \frac{\partial p_{\theta, i, j}}{\partial v_i^k}(v^k) (p_{\theta, i, j}(v^k) d_{\theta, i, j}) \quad (3.7)$$

Cette formule constitue la base de tout le calcul. Nous avons donc deux termes à déterminer : $\frac{\partial p_{\theta, i, j}}{\partial v_i}(v)$ et $p_{\theta, i, j}(v)$. Ce sont donc ces deux calculs que nous allons développer dans les parties suivantes.

3.5.4 Calcul des valeurs des cartes de projection : $p_{\theta, i, j}(v)$

Reprenons la formule à minimiser.

$$E(v) = \sum_{\theta, i, j} d_{\theta, i, j} p_{\theta, i, j}(v)^2 \quad (3.8)$$

La première étape est donc de pouvoir calculer les $p_{\theta, i, j}(v)$. Pour cela, il faut donc calculer les points d'intersection entre les droites définies par $x = i$ et $y = j$ et la forme. Considérons l'une de ses droites l . On parcourt alors l'ensemble des faces triangulaires. Soit l'une d'elles f_0 et v_0, v_1 et v_2 les points la constituant et exprimés dans l'ordre trigonométrique lorsqu'on la regarde de face et de l'extérieur et enfin \vec{n} son vecteur normal. Il s'agit donc d'une part de calculer le point d'intersection entre la droite l et le plan pf de la face, et ensuite de voir si ce point appartient ou non à cette face, soit encore si il est à l'intérieur ou non du triangle.

Note : si le plan pf est parallèle à l , ce que l'on détermina grâce à \vec{n} , on considère dans ce cas qu'il n'y a en aucun cas intersection.

Cette situation écartée, on a donc comme équation du plan pf :

$$(x - x_1) x_n + (y - y_1) y_n + (z - z_1) z_n = 0 \quad (3.9)$$

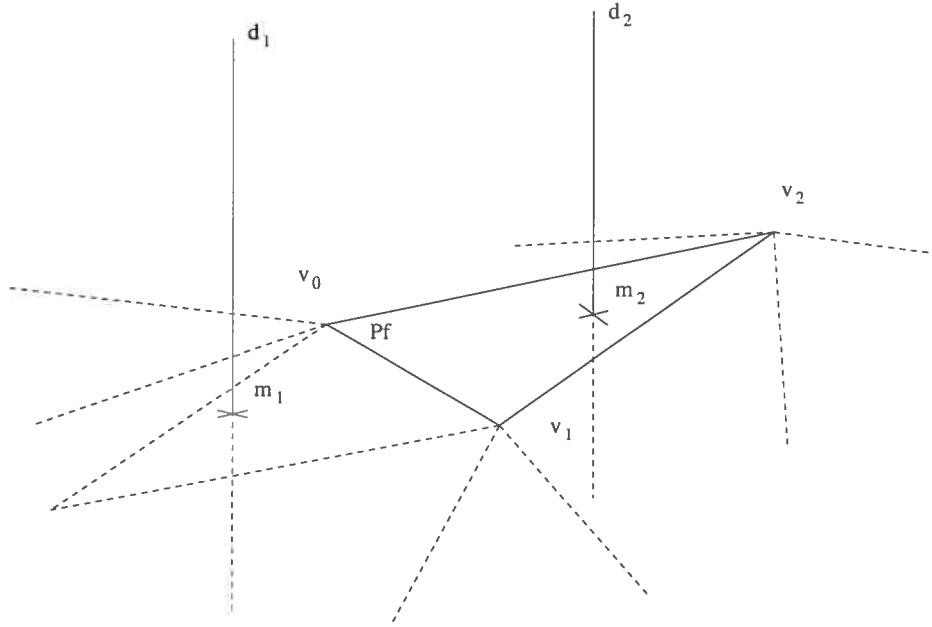


FIG. 3.9 – On voit ici l'intersection m_1 de la droite de projection d_1 avec le plan P_f formé par la facette $v_0v_1v_2$. m_1 n'appartient pas au triangle $v_0v_1v_2$ donc il sera ignoré. La droite d_2 coupe le plan P_f en m_2 qui appartient au triangle. On enregistrera donc la hauteur de ce point.

Or, au point d'intersection, $x = i$ et $y = j$, il ne nous reste que z à déterminer et l'on a :

$$z = \frac{x_1 x_n + y_1 y_n + z_1 z_n}{z_n} \quad (3.10)$$

Pour savoir ensuite si le point, que l'on note M , appartient au triangle, il faut projeter le triangle et le point M dans un plan horizontal. Appelons p_0 , p_1 et p_2 les projetés respectifs de v_0 , v_1 et v_2 . Il faut donc vérifier que M est dans le triangle $p_0p_1p_2$. Pour cela, nous allons observer le signe des deux produits scalaires suivants : $p_0\vec{p}_1 \cdot p_0\vec{M}$ $p_0\vec{p}_2 \cdot p_0\vec{M}$ et $p_1\vec{p}_2 \cdot p_1\vec{M}$ $p_1\vec{p}_0 \cdot p_1\vec{M}$.

Si ceux là sont tous les deux négatifs ou nuls, alors M appartient bien au triangle. Pour comprendre cela, regardons la figure 3.10. On voit apparaître deux

zones hachurées. M appartient au triangle si il appartient aux deux zones à la fois. Hors la zone en pointillés forts correspond à l'ensemble des points qui vérifie $p_0\vec{p}_1 \cdot p_0\vec{M} \cdot p_0\vec{p}_2 \cdot p_0\vec{M} > 0$. Cela correspond, comme on peut le voir sur la figure, à ce que les angles p_1p_0m et mp_0p_2 soient de sens opposé. De même, la deuxième zone en pointillés faibles correspond à la zone où $p_1\vec{p}_2 \cdot p_1\vec{M} \cdot p_1\vec{p}_0 \cdot p_1\vec{M} > 0$. En étudiant ses deux signes on en déduit donc bien si M appartient ou non au triangle.

On remarque que l'ensemble des trois droites formant le triangle répond à ce critère et en particulier les parties autres que les arêtes du triangle (cas où la composante est nulle). Il faut donc vérifier dans ce cas que le point est bien sur l'arête et non sur sa prolongation.

Lorsque la droite $l_{i,j}$ traverse effectivement la face, il reste à savoir si elle entre ou sort de la forme, pour cela il suffit d'observer le signe de la coordonnée n_z . En notant s_m ce signe où m correspond au numéro de la face en question, on a alors :

$$p_{i,j}(v) = \sum_m s_m z_m \quad (3.11)$$

Notons que pour l'implémentation, il faut veiller à ce que lorsque l'intersection se fait sur une arête d'une face, il ne faut compter qu'une seule fois cette intersection, c'est-à-dire ne pas la recompter avec les autres faces partageant la même arête. De plus, pour savoir si une intersection a déjà été calculée, on compare la valeur obtenue avec celle déjà enregistrée. Or, pour connaître le point d'une droite, on se sert de points de cette droite, et si mathématiquement le résultat est le même quelque soient les points choisis pour représenter la droite, la valeur informatique et donc approchée, variera. Il s'agit donc de mettre en place certaines règles qui

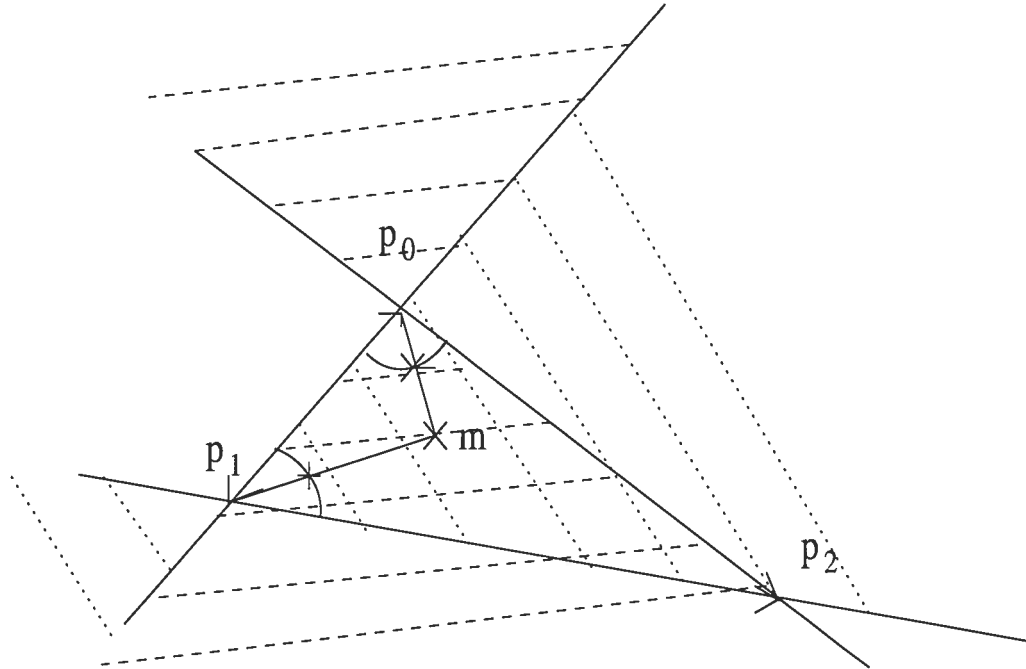


FIG. 3.10 – Représentation du triangle $p_0p_1p_2$ et d'un point M à l'intérieur de celui-ci. La zone en pointillés forts correspond à l'ensemble des points qui vérifie $p_0\vec{p}_1 \cdot p_0\vec{M} \cdot p_0\vec{p}_2 \cdot p_0\vec{M} = 0$. La deuxième zone en pointillés faibles correspond à la zone où $p_1\vec{p}_2 \cdot p_1\vec{M} \cdot p_1\vec{p}_0 \cdot p_1\vec{M} = 0$.

imposent à chaque fois un choix unique des points dans les équations (on se basera dans notre code sur l'ordre des ordonnées, puis des abscisses).

3.5.5 Calcul de $\frac{\partial p_{\theta,i,j}}{\partial v_i}(v)$ présent dans l'expression du gradient

Il nous faut alors calculer $\frac{\partial p_{\theta,i,j}}{\partial v_i}(v)$. Ce calcul est complexe et constitue la partie longue et délicate de l'algorithme. Nous allons voir qu'il faut tout d'abord bien isoler le problème en restreignant l'étude non pas sur l'ensemble de notre forme mais sur des volumes bien spécifiques, c'est-à-dire restreints à un petit sous-volume du volume total. Nous chercherons ensuite à diviser ces volumes en plusieurs vo-

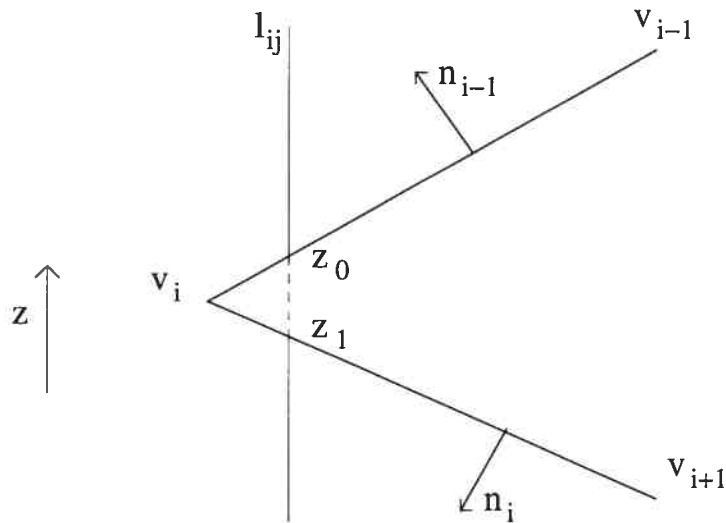


FIG. 3.11 – Exemple en 2 dimensions d'une droite $l_{i,j}$ coupant la figure en deux hauteurs z_0 et z_1 . n_{i-1} est orienté vers le haut et n_i vers le bas donc on aura $p_{i,j}(v) = z_0 - z_1$ ce qui correspond bien à la distance traversée.

lumes simples dans le but encore une fois de découper le problème et de rendre les calculs plus accessibles. Enfin dans ce contexte rendu plus simple et plus clair, nous exprimerons $p_{0,i,j}(v)$ puis sa dérivée $\frac{\partial p_{0,i,j}}{\partial v_i}(v)$.

3.5.5.1 Restriction du volume à considérer

Commençons donc par restreindre le volume d'étude afin de rendre le problème plus clair et plus simple. Comme nous l'avons vu, le gradient a pour composante l'ensemble des dérivées par rapport à l'ensemble des points de la figure. Focalisons-nous alors sur une seule de ces dérivées autour du point v_i . Si on imagine l'ensemble de la forme, on comprend que si on bouge un de ses points, ce point ne va agir que sur les arêtes le reliant à ses voisins. Donc si l'on considère la forme complète comme étant l'addition du volume constitué par le point v_i et ses voisins que l'on notera V_i , et le volume constitué par tous les points sauf v_i , on constate alors que

seul le volume V_i varie lorsque l'on déplace le point v_i .

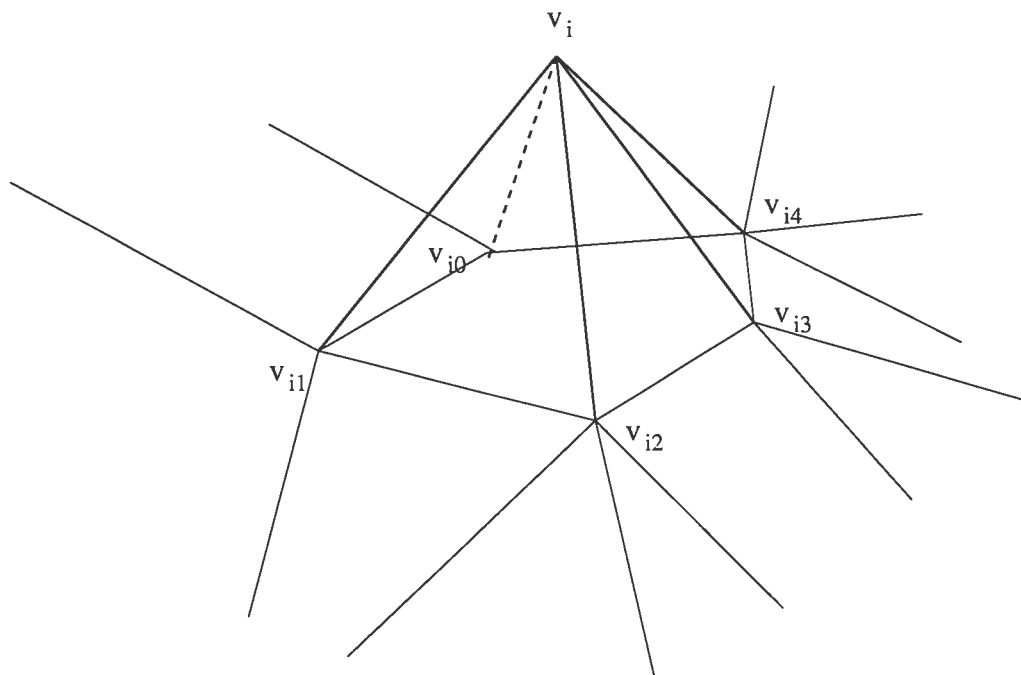


FIG. 3.12 – Volume formé par le point v_i et ses voisins que l'on isole du reste de la forme.

Autrement dit lorsque l'on calcule la dérivée partielle $\frac{\partial p}{\partial v_i}(v)$, en notant v_{-v_i} l'ensemble des points excepté v_i , v_{v_i} les points v_i et ses voisins et dv et dv_{v_i} traduisant le petit déplacement de v_i , on peut écrire :

$$\frac{\partial p}{\partial v_i}(v) = p(v + dv) - p(v) = p(v_{-v_i}) + p(v_{v_i} + dv_{v_i}) - (p(v_{-v_i}) + p(v_{v_i})) \quad (3.12)$$

Soit :

$$\frac{\partial p}{\partial v_i}(v) = \frac{\partial p}{\partial v_i}(v_{v_i}) \quad (3.13)$$

3.5.5.2 Découpage en volumes élémentaires

On s'intéresse dès lors uniquement au volume V_i . On cherche alors à diviser ce volume en volumes élémentaires, c'est-à-dire des volumes constitués uniquement de quatre points. Quelque soit le nombre de points considérés au départ, il est toujours possible de diviser un volume en volumes élémentaires, (voir figure 3.13), cependant il est difficile d'établir un découpage général dès lors que le nombre de points varie ainsi que leur position.

3.5.5.2.1 Calcul d'un point supplémentaire : e . On décide alors de rajouter en permanence un point supplémentaire que l'on note e , qui dépend du barycentre des voisins de v_i , et qui permet un découpage général du volume : on prend v_i , deux voisins consécutifs de v_i : v_{i1} et v_{i2} , et e , et de plus permet de rendre le volume moins plat et donc d'avoir des valeurs moins extrêmes lors des calculs de dérivations.

Détaillons le calcul de ce point e . Il faut dans un premier temps calculer le barycentre des points voisins de v_i que l'on note : bary . Ses coordonnées sont donc les moyennes des coordonnées des points voisins. On cherche ensuite à obtenir un vecteur que l'on appellera vecteur normal et extérieur au volume V_i et que l'on note \vec{n}_i . \vec{n}_i se définit alors de la façon suivante :

$$\vec{n}_i = c_i \left(\text{bary}_{\text{voisins de } v_i} \vec{v}_{i1} - \text{bary}_{\text{voisins de } v_i} \vec{v}_{i2} \right) \quad (3.14)$$

Où c_i est un coefficient servant à normaliser le vecteur n_i . Il est donc propre au vecteur \vec{n}_i et vaut l'inverse de la norme du vecteur $\text{voisins de } v_i \text{ bary}_{\text{voisins de } v_i} \vec{v}_{i1} - \text{bary}_{\text{voisins de } v_i} \vec{v}_{i2}$.

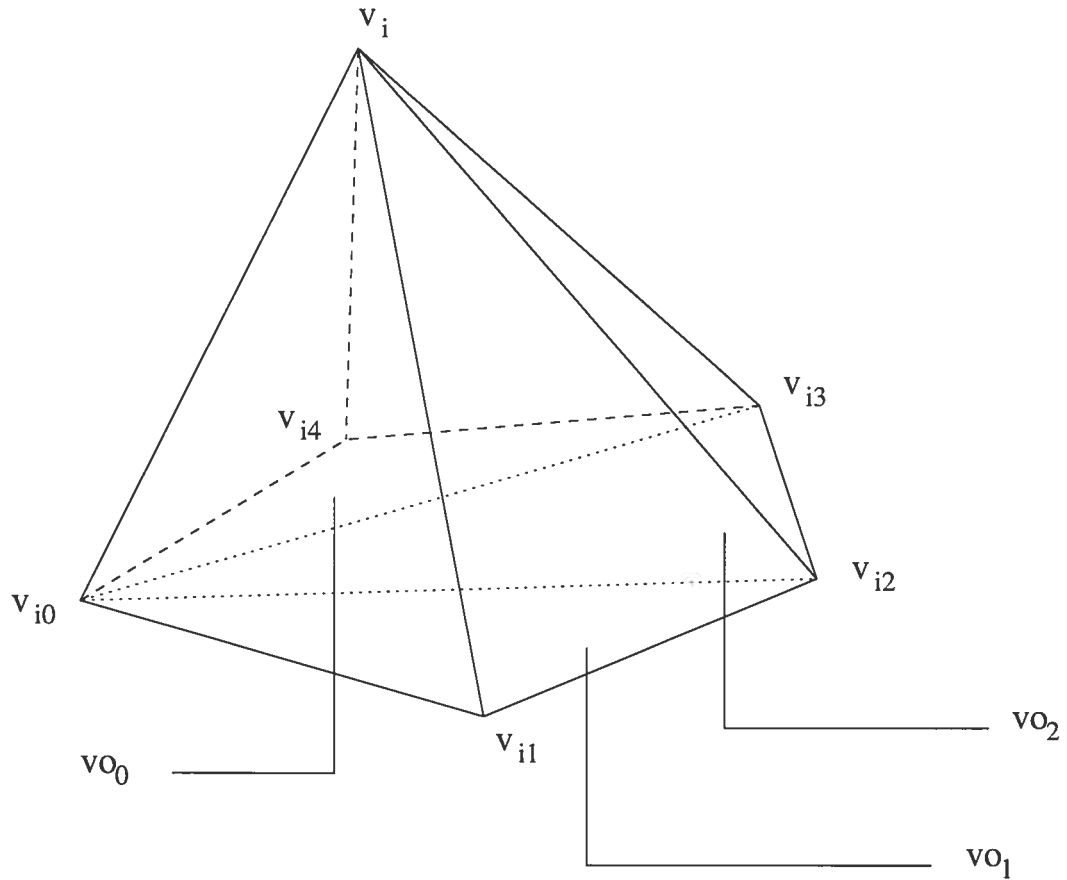


FIG. 3.13 – Découpage d'un volume défini avec les voisins du point v_i en 3 volumes élémentaires v_{o0} , v_{o1} et v_{o2} . (Soit encore les volumes formés par les points $v_i v_{i0} v_{i3} v_{i4}$, $v_i v_{i0} v_{i1} v_{i2}$ et $v_i v_{i0} v_{i2} v_{i3}$.)

Enfin on calcule un ensemble de produits scalaires ps_j définis par :

$$ps_j = v_i \overline{\text{bary}} v_i \vec{v}_j \quad (3.15)$$

où j parcourt l'ensemble des voisins de v_i . Dans le cas le plus général, on aura tous les ps_j positifs, et on définit alors le point e tel que :

$$e = v_i + \max \left(1 - \frac{ps_j}{v_i \overline{\text{bary}}^2} \right) v_i \overline{\text{bary}} \quad (3.16)$$

Sinon, on calcule un scalaire α ,

$$\alpha = 2 \min \left(\vec{n}_i \cdot \vec{bary}v_i, \vec{n}_i \cdot \vec{bary}v_{i1}, \dots, \vec{n}_i \cdot \vec{bary}v_{in} \right) \quad (3.17)$$

Et e est alors égal à :

$$e = \vec{bary} + \alpha \varepsilon \vec{n}_i \quad (3.18)$$

Avec ε égal au signe de $\vec{bary} \cdot \vec{n}_i$. Ce dernier cas permet à e d'être bien en dehors du volume de base.

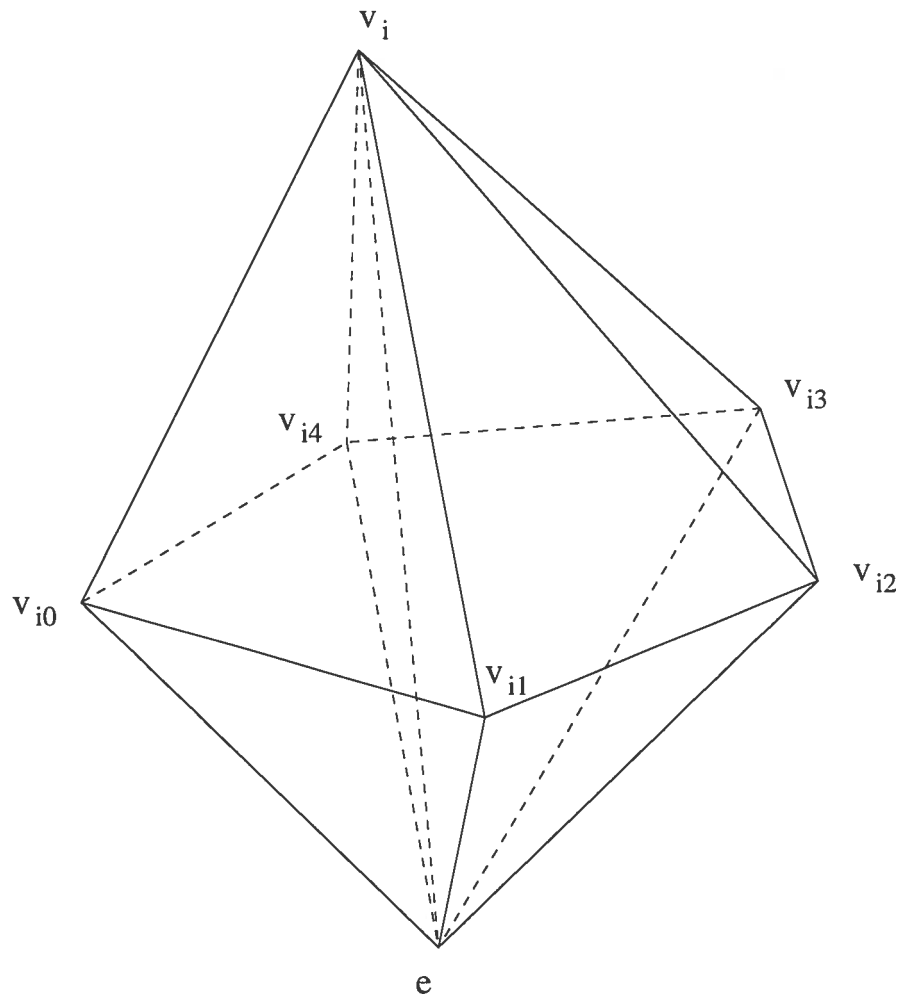


FIG. 3.14 – Ajout d'un point e . On crée ici les volumes : $v_i v_{i0} e v_{i1}$, $v_i v_{i1} e v_{i2}$, $v_i v_{i2} e v_{i3}$, $v_i v_{i3} e v_{i4}$ et $v_i v_{i4} e v_{i0}$.

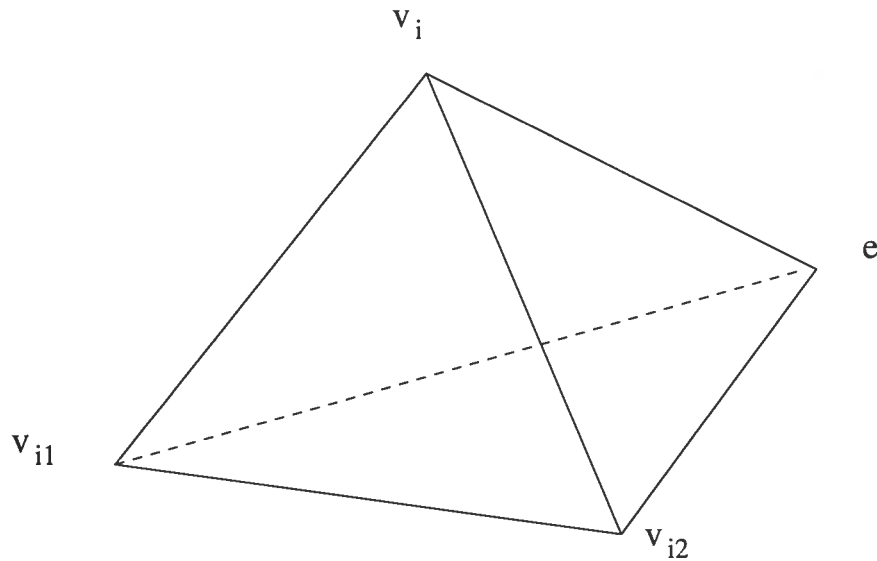


FIG. 3.15 – Volume élémentaire constitué du point v_i , de deux de ses voisins v_{i1} et v_{i2} et du point e .

3.5.5.3 Définition de trois cas à distinguer pour les calculs

On considère à présent, un volume élémentaire constitué de v_i , de deux de ses voisins v_{i1} , v_{i2} et de e qu'on a à présent défini.

Bien que ce volume soit élémentaire, celui-ci est encore trop complexe pour effectuer simplement les calculs de dérivation. Il va donc nous falloir rediviser ce volume en d'autres volumes élémentaires toujours de quatre points avec une base triangulaire et un axe vertical comme illustré figure 3.16.

Pour pouvoir effectuer ce découpage, il faut distinguer plusieurs cas en fonction de la projection du volume élémentaire selon l'axe des lignes de projection, c'est-à-dire selon l'axe (O, z) . On réalise cette projection et on distingue alors trois cas. En notant p_0 le projeté de v_i et p_1 , p_2 et p_3 les projetés des trois autres points, on a donc :

1. un triangle avec p_0 à l'intérieur du triangle.
2. un triangle avec p_0 faisant partie de ce triangle.
3. un losange.

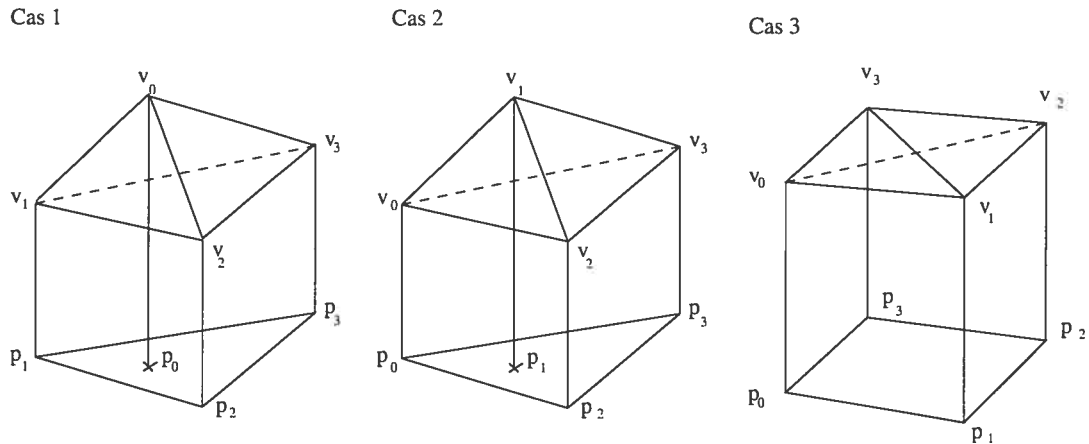


FIG. 3.16 – Illustration des trois cas en fonction de la projection du volume élémentaire suivant l'axe (O, z) .

On rappelle que le but final de ces calculs est de déterminer les dérivés des projections, c'est-à-dire la valeur de dl lorsque l'on déplace v_i selon x, y ou z , comme l'illustre la figure 3.17.

3.5.5.4 Découpage en sous-volumes élémentaires

Dans les deux premiers cas, on crée un quatrième point p'_0 qui est le projeté du point du centre sur le plan défini par les trois autres formant le triangle. On crée ainsi trois nouveaux volumes élémentaires autour de l'axe défini par p'_0 et le point central, et chaque droite de projection ne coupe qu'un seul de ces volumes. Dans le troisième cas, si on suppose que l'ordre des points du quadrilatère est v_1, v_2, v_0 et v_3 , on va alors créer deux nouveaux points p_4 et p'_4 . Ces deux points

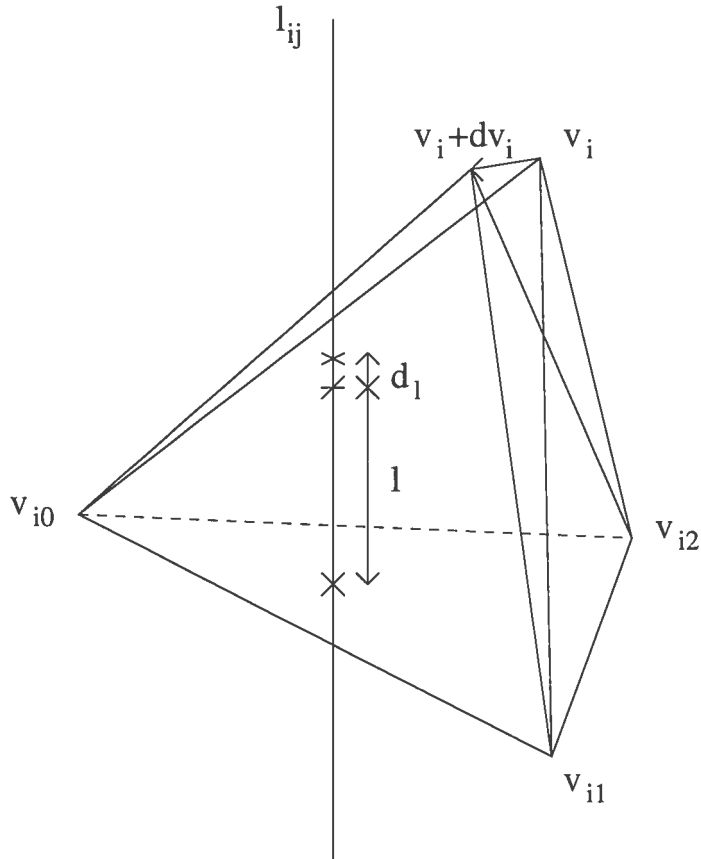


FIG. 3.17 – On fait apparaître sur cette figure un déplacement de v_i qui entraîne ainsi une variation d_l de la distance l traversée par la droite $l_{i,j}$ à travers le volume.

appartiennent à l'intersection des deux plans verticaux passant l'un par la droite (v_3, v_2) , l'autre par la droite (v_1, v_0) et p_4 appartient à la droite (v_1, v_0) et p'_4 à la droite (v_2, v_3) . On a alors aussi quatre volumes élémentaires autour de l'axe $[p_4, p'_4]$.

3.5.5.5 Calcul des $p(i, j)$

Ayant bien réduit le volume à étudier et décomposé celui-ci, nous pouvons à présent calculer l'expression de $p(i, j)$. On se place dans un volume élémentaire

constitué des points p_0, p'_0 et de deux voisins de v_0 , ou de p_4, p'_4 et de deux voisins de v_0 . Seul le nom des points changeant on prendra ici comme nom : p_0, p'_0, p_1 et p_2 . Pour simplifier les calculs, nous allons mettre p_1, p_2 et p'_0 sur le même plan et placer p_0 à la distance P au dessus de p'_0 . Les valeurs des projections, et par conséquent leurs dérivées également, restent parfaitement identiques à celles du volume original. Intuitivement, ce n'est qu'une sorte de projection sur un plan tout en conservant le volume. Pour s'en convaincre, une simple application du théorème de Thales nous le démontre. Considérons la figure 3.18 :

On considère que le volume initial est formé des points p_0, p'_0, p'_1 et p'_2 . Pour obtenir p'_0, p_1 et p_2 sur un plan horizontal, on a déplacé p'_1 et p'_2 à hauteur de p'_0 ce qui donne donc respectivement les points p_1 et p_2 . r'_1 et r'_2 sont les deux points d'intersection entre le volume initial et la droite de projection et r_1 et r_2 sont ceux entre le volume déformé et la droite de projection. r'_3 est le point d'intersection entre $(p'_1 r'_1)$ et $(p_0 p'_2)$, r'_4 entre $(p'_1 r'_2)$ et $(p'_0 p'_2)$, r_3 entre $(p_1 r_2)$ et $(p_0 p_2)$ et r_4 entre $(p_1 r_3)$ et $(p'_0 p_2)$. $p_1(i, j) = r'_1 r'_2$ représente la projection obtenue avec le volume initial et $p_2(i, j) = r_1 r_2$ celle obtenue avec le volume déformé. Il s'agit donc de montrer que $p_1(i, j) = p_2(i, j)$. Pour cela montrons que de façon générale, pour tout triangle ABC , tel que BC soit égale à b et de hauteur issue de A égale à h . Si D et E sont deux points respectifs de (A, B) et (A, C) tels que (D, E) soit parallèle à (B, C) et située à une distance d de celle-ci, alors la distance DE est égale à une constance c . Prenons un tel triangle et traçons sa hauteur. Soit H le projeté de A sur (B, C) et H_1 l'intersection entre (A, H) et (D, E) tels que représenté sur la figure 3.19.

D'après le théorème de Thales, sur la figure 3.19, dans le triangle ABC , on a $\frac{AD}{AB} = \frac{DE}{BC}$ et dans le triangle AHB , on a $\frac{AH_1}{AH} = \frac{AD}{AB}$. Donc $\frac{DE}{BC} = \frac{AH_1}{AH}$ soit encore

$$DE = AH_1 \cdot \frac{BC}{AH} = c.$$

Il ne nous reste plus qu'à appliquer cette propriété dans notre figure 3.18. r_1 , r'_1 , r_2 , r'_2 sont sur la même droite verticale et p_1 et p'_1 sont également sur une droite verticale, donc ces six points sont sur un même plan vertical. On en déduit par construction que r_3 , r'_3 , r_4 , r'_4 sont également dans ce même plan. De plus (p_0, p'_0) et (p_2, p'_2) sont deux droites verticales donc ces quatre points sont dans un même plan vertical et par construction r_3 , r'_3 , r_4 , r'_4 sont aussi dans ce plan. Ces quatre derniers étant à l'intersection de deux plans verticaux sont donc sur une même droite verticale. D'après la propriété démontrée, on en déduit que dans les triangles $p_0p'_2p'_0$ et $p_0p_2p'_0$, $r_3r_4 = r'_3r'_4$. De ce fait, on peut appliquer également cette propriété aux triangles $r'_3r_4p'_1$ et $r_3r_4p_1$ et on en déduit que $r_1r_2 = r'_1r'_2$ c'est-à-dire que les valeurs des projections sont identiques.

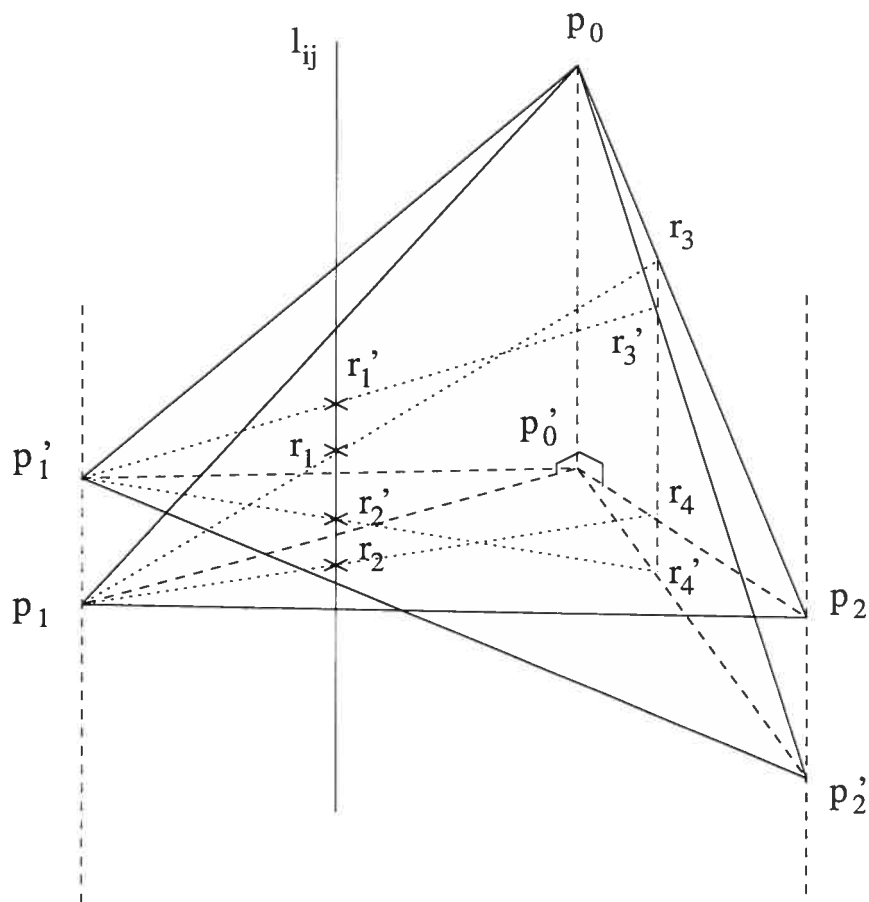


FIG. 3.18 – Schéma représentant le volume d'origine formé des points p_0, p_0', p_1' et p_2' et celui modifié avec les points p_0, p_0', p_1 et p_2 . On fait apparaître les points d'intersections entre la droite de projection et le volume d'origine, soit : r_1' et r_2' et ceux avec le volume modifié : r_1 et r_2 .

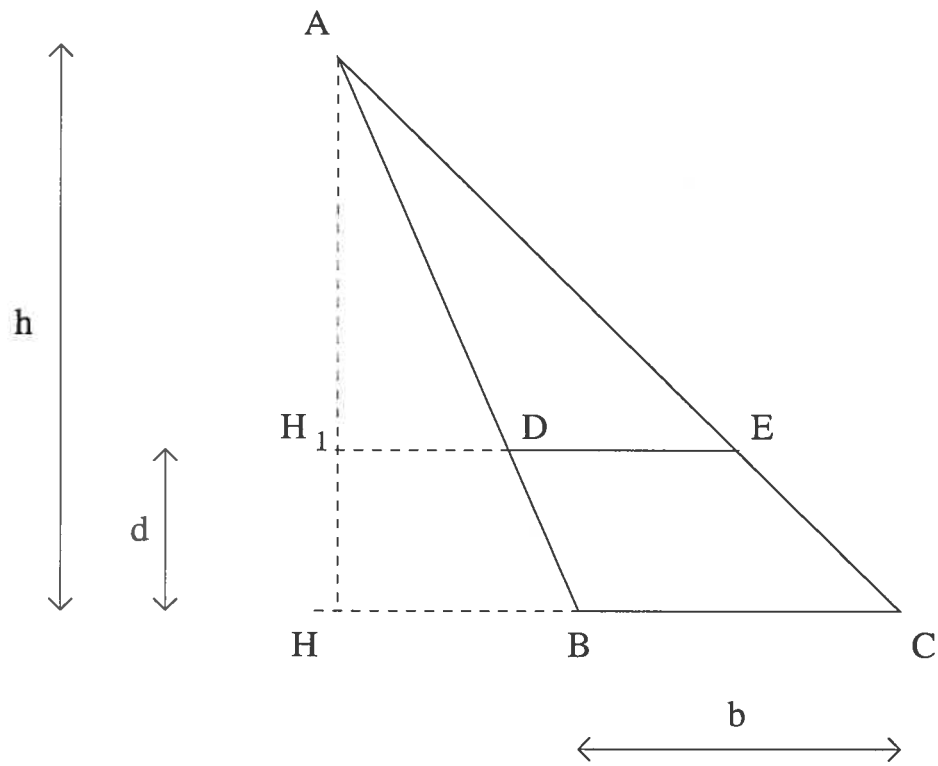


FIG. 3.19 – Figure illustrant l'invariance de la longueur DE pour des d , h et b constants.

On peut donc considérer dorénavant pour nos calculs le volume p_0, p'_0, p_1, p_2 avec p'_0, p_1 et p_2 dans le même plan horizontal qu'on considérera à la hauteur $h = 0$. Pour connaître la valeur de la projection $p(i, j)$, il faut donc connaître la hauteur du point d'intersection entre la droite d'équation $x = i$ et $y = j$ et le plan (p_0, p_1, p_2) . Si \vec{n} est un vecteur normal à ce plan et M un point de ce plan, celui-ci peut être défini par l'équation :

$$p_1 \vec{M} \cdot \vec{n} = 0 \quad (3.19)$$

Soit :

$$(x - x_1) x_n + (y - y_1) y_n + (z - z_1) z_n = 0 \quad (3.20)$$

Et $z_1 = 0$ donc

$$z = \frac{(x_1 - x) x_n + (y_1 - y) y_n}{z_n} \quad (3.21)$$

On prend pour $\vec{n}, \vec{n} = p_1 \vec{p}_0 - p_1 \vec{p}_2$ soit :

$$\vec{n} = \begin{pmatrix} P(y_2 - y_1) \\ P(x_2 - x_1) \\ (x_0 - x_1)(y_2 - y_1) - (y_0 - y_1)(x_2 - x_1) \end{pmatrix} \quad (3.22)$$

D'où en remplaçant dans l'équation précédente, on a :

$$z = P \cdot \frac{(x - x_1) \cdot (y_1 - y_2) + (y - y_1) \cdot (x_2 - x_1)}{(x_1 - x_0) \cdot (y_2 - y_0) - (y_1 - y_0) \cdot (x_2 - x_0)} \quad (3.23)$$

Soit

$$p(i, j) = P \cdot \frac{(i - x_1) \cdot (y_1 - y_2) + (j - y_1) \cdot (x_2 - x_1)}{(x_1 - x_0) \cdot (y_2 - y_0) - (y_1 - y_0) \cdot (x_2 - x_0)} \quad (3.24)$$

Nous avons donc exprimé $p(i, j)$ en fonction de P . Il nous faut donc exprimer P à présent avant de calculer finalement la dérivée : $\frac{\partial p}{\partial v_i}(i, j)$.

3.5.5.6 Calcul de la hauteur des axes : P

Plaçons-nous dans l'un des deux premiers cas. Afin de généraliser l'écriture on prend comme nom pour les points : p_0 pour le point central et p_1, p_2 et p_3 pour les points formants le triangle et dans un ordre trigonométrique. On garde p'_0 tel qu'on l'a défini.

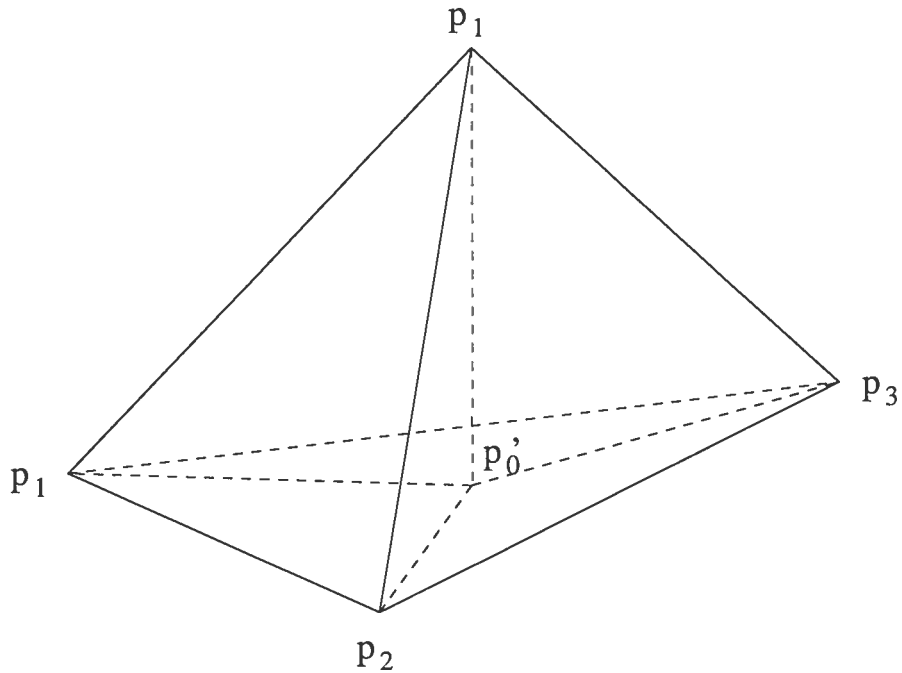


FIG. 3.20 – Schéma d'un volume élémentaire (dans le cas où sa projection forme un triangle) divisé en trois volumes avec l'ajout du point p'_0 .

Appelons P la distance entre p_0 et p'_0 . On a alors $P = z'_0 - z_0$ p'_0 est le point d'intersection entre la droite d'équation $x = x_0$ et $y = y_0$ et le plan (p_1, p_2, p_3) . Si \vec{n} est un vecteur normal à ce plan, p'_0 vérifie alors l'équation :

$$\vec{p_1 p'_0} \cdot \vec{n} = 0 \quad (3.25)$$

Soit :

$$(z'_0 \quad z_1) \cdot z_n + (y_0 \quad y_1) \cdot y_n + (x_0 \quad x_1) \cdot x_n = 0 \quad (3.26)$$

Et sachant que $y'_0 = y_0$ et $z'_0 = z_0$, on a donc :

$$z'_0 = \frac{(x_1 \quad x_0) \cdot x_n + (y_1 \quad y_0) \cdot y_n}{z_n} + z_1 \quad (3.27)$$

Avec z_n non nul. Et finalement :

$$z'_0 \quad z_0 = \frac{(x_1 \quad x_0) \cdot x_n + (y_1 \quad y_0) \cdot y_n + (z_1 \quad z_0) \cdot z_n}{z_n} \quad (3.28)$$

Pour \vec{n} on peut prendre $\vec{n} = p_1\vec{p}_2 - p_1\vec{p}_3$, et donc :

$$x_n = (y_2 \quad y_1) \cdot (z_3 \quad z_1) - (z_2 \quad z_1) \cdot (y_3 \quad y_1); \quad (3.29)$$

$$y_n = (z_2 \quad z_1) \cdot (x_3 \quad x_1) - (x_2 \quad x_1) \cdot (z_3 \quad z_1); \quad (3.30)$$

$$z_n = (x_2 \quad x_1) \cdot (y_3 \quad y_1) - (y_2 \quad y_1) \cdot (x_3 \quad x_1); \quad (3.31)$$

Remarquons que dans le code, on vérifie que z_n n'est pas nul. z_n égal à 0 signifie que p_1p_2 et p_1p_3 sont colinéaires. Or, ce cas est d'une part presque irréalisable dans la pratique. L'un des trois points est rappelons le, le barycentre d'un ensemble de points, il faut donc une disposition peu « naturelle » des points voisins pour que celui-ci soit aligné avec deux de ces points consécutifs. De plus, même si mathématiquement il doit y avoir l'égalité, l'approximation informatique des nombres réduit fortement cette possibilité. Enfin, si cela se réalise, cela correspond donc à un volume nul (trois points alignés et un quatrième forment un plan et non un volume) et on n'a de toutes les manières pas à traiter ce cas.

Plaçons-nous à présent dans le troisième cas, celui où les quatre points projetés forment un quadrilatère.

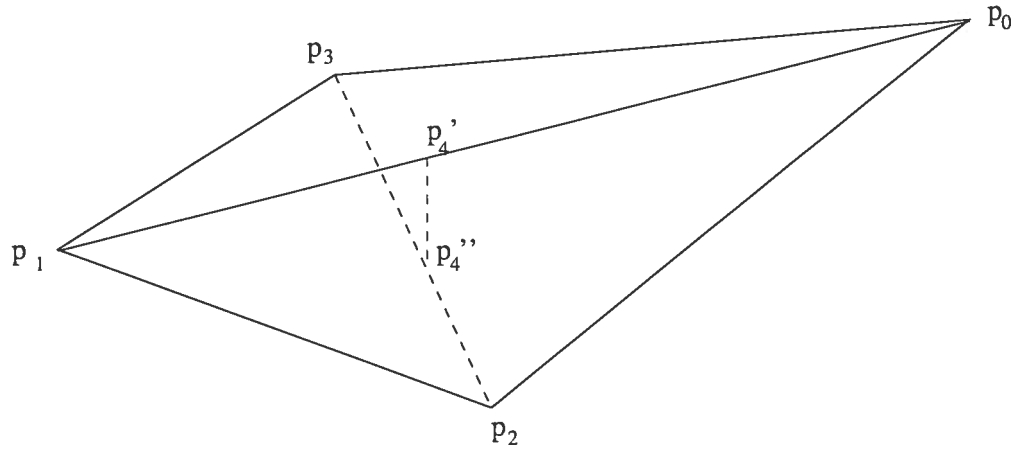


FIG. 3.21 – Schéma d'un volume élémentaire (dans le cas où sa projection forme un quadrilatère) divisé en quatre volumes avec l'ajout des points p_4' et p_4'' .

Il s'agit alors de déterminer z_4 et z_4' . Commençons par z_4 . Il faut au préalable calculer x_4 et y_4 (qui sont respectivement égaux à x_4' et y_4'). Pour cela on écrit les équations des droites (p_1, p_0) et (p_2, p_3) projetées dans le plan (O, x, y) . On a alors :

$$(y - y_1) \cdot \left(\frac{x_0 - x_1}{y_0 - y_1} \right) + x_1 = x \quad (3.32)$$

$$(y - y_3) \cdot \left(\frac{x_2 - x_3}{y_2 - y_3} \right) + x_3 = x \quad (3.33)$$

On met bout à bout les deux égalités et on en déduit :

$$(y - y_1) \cdot \left(\frac{x_0 - x_1}{y_0 - y_1} \right) + x_1 = (y - y_3) \cdot \left(\frac{x_2 - x_3}{y_2 - y_3} \right) + x_3 \quad (3.34)$$

Et après développement et simplification, on obtient :

$$y_4 = \frac{y_0 \cdot (y_2 \cdot (x_3 \ x_1) + y_3 \cdot (x_1 \ x_2)) + x_0 \cdot y_1 \cdot (y_2 \ y_3) + y_1 \cdot (y_3 \cdot x_2 \ x_3 \cdot y_2)}{(x_0 \ x_1) \cdot (y_2 \ y_3) \ (x_2 \ x_3) \cdot (y_0 \ y_1)} \quad (3.35)$$

On fait de même pour trouver x_4 en posant comme équation des 2 droites :

$$(x \ x_1) \cdot \left(\frac{y_0 \ y_1}{x_0 \ x_1} \right) + y_1 = y \quad (3.36)$$

$$(x \ x_3) \cdot \left(\frac{y_2 \ y_3}{x_2 \ x_3} \right) + y_3 = y \quad (3.37)$$

On trouve au final une expression similaire à y_4 soit :

$$x_4 = \frac{x_0 \cdot (x_2 \cdot (y_3 \ y_1) + x_3 \cdot (y_1 \ y_2)) + y_0 \cdot x_1 \cdot (x_2 \ x_3) + x_1 \cdot (x_3 \cdot y_2 \ y_3 \cdot x_2)}{(y_0 \ y_1) \cdot (x_2 \ x_3) \ (y_2 \ y_3) \cdot (x_0 \ x_1)} \quad (3.38)$$

On remarque que pour les raisons déjà expliquées lors du cas précédent, les dénominateurs de x_4 et y_4 sont non nuls. On peut alors écrire des équations paramétriques de la droite (p_1, p_0) en posant que :

$$p_1 \vec{p}_0 \quad p_1 \vec{m} = \vec{0} \quad (3.39)$$

Soit :

$$(y_0 \ y_1) \cdot (z \ z_1) \quad (z_0 \ z_1) \cdot (y \ y_1) = 0 \quad (3.40)$$

$$(z_0 \ z_1) \cdot (x \ x_1) \quad (x_0 \ x_1) \cdot (z \ z_1) = 0 \quad (3.41)$$

$$(x_0 \ x_1) \cdot (y \ y_1) \quad (y_0 \ y_1) \cdot (x \ x_1) = 0 \quad (3.42)$$

Et donc en z_4 on peut extraire les deux égalités :

$$(y_0 \ y_1) \cdot (z_4 \ z_1) \quad (z_0 \ z_1) \cdot (y_4 \ y_1) = 0 \quad (3.43)$$

$$(z_0 \ z_1) \cdot (x_4 \ x_1) - (x_0 \ x_1) \cdot (z_4 \ z_1) = 0 \quad (3.44)$$

Si $y_0 = y_1$ on peut alors se servir de l'équation $(y_0 \ y_1) \cdot (z_4 \ z_1) - (z_0 \ z_1) \cdot (y_4 \ y_1) = 0$ et on obtient :

$$z_4 = \frac{(z_0 \ z_1) \cdot (y_4 \ y_1)}{(y_0 \ y_1)} + z_1 \quad (3.45)$$

Sinon on se sert de l'équation $(z_0 \ z_1) \cdot (x_4 \ x_1) - (x_0 \ x_1) \cdot (z_4 \ z_1) = 0$ et on a

$$z_4 = \frac{(z_0 \ z_1) \cdot (x_4 \ x_1)}{(x_0 \ x_1)} + z_1 \quad (3.46)$$

Encore une fois, on ne peut avoir à la fois $y_0 = y_1$ et $x_0 = x_1$. On procède de la même façon pour z'_4 en se servant de l'équation $p_2 \vec{p}_3 - p_2 \vec{m} = 0$ et on trouve

$$z'_4 = \frac{(z_3 \ z_2) \cdot (y_4 \ y_2)}{(y_3 \ y_2)} + z_2 \quad (3.47)$$

ou

$$z'_4 = \frac{(z_3 \ z_2) \cdot (x_4 \ x_2)}{(x_3 \ x_2)} + z_2 \quad (3.48)$$

On en déduit $P = \text{signe} \cdot (z_4 \ z'_4)$ avec $\text{signe} = 1$ ou $\text{signe} = -1$ de sorte que P soit positif.

3.5.5.7 Calcul des $\frac{\partial p}{\partial v_i}(i, j)$

Nous arrivons à la phase finale du calcul. Nous avons trouvé précédemment que $p(i, j)$ était de la forme : $p(i, j) = P \cdot A(i, j)$ où A est une fonction. On en déduit que $\frac{\partial p}{\partial v_i}(i, j)$ est de la forme :

$$\frac{\partial p}{\partial v_i}(i, j) = \frac{\partial P}{\partial v_i} \cdot A(i, j) + P \cdot \frac{\partial A}{\partial v_i}(i, j) \quad (3.49)$$

Il nous reste donc à calculer $\frac{\partial P}{\partial v_i}$ et $\frac{\partial A}{\partial v_i}(i, j)$.

3.5.5.7.1 Calcul des $\frac{\partial P}{\partial v_0}$. Prenons les 3 cas de disposition de p_0 par rapport aux trois autres, un par un.

1er cas, p_0 est à l'intérieur du triangle formé par les trois autres.

L'axe vertical est donc $[p_0 p'_0]$. Les 3 volumes existants sont $p_0 p'_0 p_1 p_2$, $p_0 p'_0 p_2 p_3$ et $p_0 p'_0 p_3 p_1$ (revoir figure 3.20). Lors de l'implémentation afin de ne pas écrire une formule par volume, il convient juste de changer les noms des deux points autres que p_0 et p'_0 . On travaille donc toujours avec un volume dont les points sont $p_0 p'_0 p_1 p_2$ et on dérive par rapport à p_0 (voir figure 3.22).

Reprenons la formule de P dans le cas étudié. On a

$$P = \frac{(x_1 - x_0) \cdot x_n + (y_1 - y_0) \cdot y_n + (z_1 - z_0) \cdot z_n}{z_n} \quad (3.50)$$

Ou encore

$$P = \text{signe} \frac{(x_1 - x_0) \cdot x_n + (y_1 - y_0) \cdot y_n + (z_1 - z_0) \cdot z_n}{z_n} \quad (3.51)$$

Où signe =1 ou -1 de façon à ce que P soit toujours positif. On a alors :

$$\frac{\partial p}{\partial v_0} = \frac{\partial p}{\partial x_0} \vec{x} + \frac{\partial p}{\partial y_0} \vec{y} + \frac{\partial p}{\partial z_0} \vec{k} \quad (3.52)$$

Soit encore :

$$\frac{\partial p}{\partial v_0} = \text{signe} \cdot \left(\frac{x_n}{z_n} \vec{i} + \frac{y_n}{z_n} \vec{j} + \vec{k} \right) \quad (3.53)$$

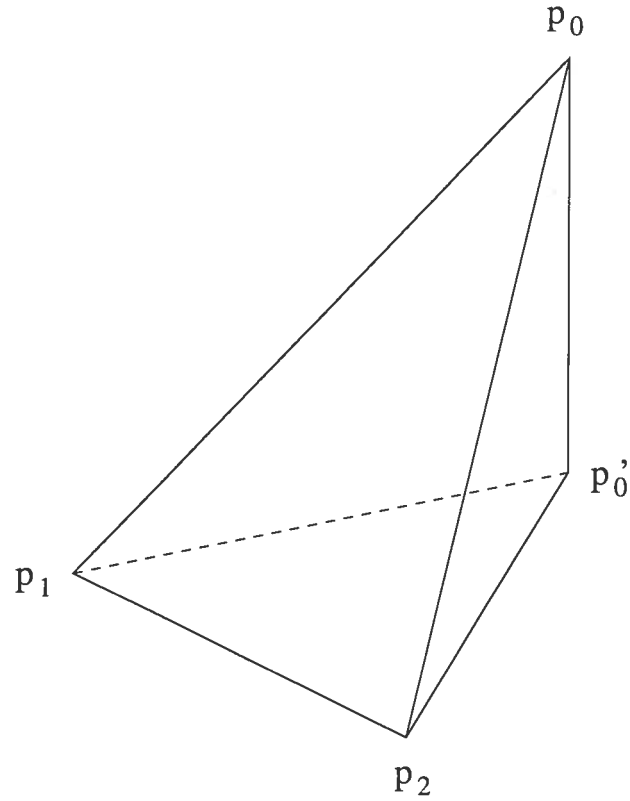


FIG. 3.22 – Volume élémentaire sur lequel on effectue les calculs.

En rappelant que \vec{n} ne dépend pas de v_0 et que

$$x_n = (y_2 \ y_1) \cdot (z_3 \ z_1) - (z_2 \ z_1) \cdot (y_3 \ y_1); \quad (3.54)$$

$$y_n = (z_2 \ z_1) \cdot (x_3 \ x_1) - (x_2 \ x_1) \cdot (z_3 \ z_1); \quad (3.55)$$

$$z_n = (x_2 \ x_1) \cdot (y_3 \ y_1) - (y_2 \ y_1) \cdot (x_3 \ x_1); \quad (3.56)$$

2ième cas, p_0 est l'un des trois points du triangle. On reprend la figure précédente et on dérive alors par rapport à p_2 (figure 3.22). Encore une fois, il suffit lors de l'implémentation d'attribuer les bons noms de points. On reprend donc

l'expression de P :

$$P = \text{signe} \frac{(x_1 \ x_0) \cdot x_n + (y_1 \ y_0) \cdot y_n + (z_1 \ z_0) \cdot z_n}{z_n} \quad (3.57)$$

On dérive par rapport à p_2 et on a :

$$\frac{\partial p}{\partial v_2} = \frac{\partial p}{\partial x_2} \vec{x} + \frac{\partial p}{\partial y_2} \vec{j} + \frac{\partial p}{\partial z_2} \vec{k} \quad (3.58)$$

Soit :

$$\begin{aligned} \frac{\partial p}{\partial v_2} &= \text{signe} \cdot \frac{z_n \cdot (x_1 \ x_0) \cdot \frac{\partial x_n}{\partial x_2} + (x_1 \ x_0) \cdot x_n \cdot \frac{\partial z_n}{\partial x_2} + z_n \cdot (y_1 \ y_0) \cdot \frac{\partial y_n}{\partial x_2} + (y_1 \ y_0) \cdot y_n \cdot \frac{\partial z_n}{\partial x_2}}{z_n^2} \vec{i} \\ &+ \text{signe} \cdot \frac{z_n \cdot (x_1 \ x_0) \cdot \frac{\partial x_n}{\partial y_2} + (x_1 \ x_0) \cdot x_n \cdot \frac{\partial z_n}{\partial y_2} + z_n \cdot (y_1 \ y_0) \cdot \frac{\partial y_n}{\partial y_2} + (y_1 \ y_0) \cdot y_n \cdot \frac{\partial z_n}{\partial y_2}}{z_n^2} \vec{j} \\ &+ \text{signe} \cdot \frac{z_n \cdot (x_1 \ x_0) \cdot \frac{\partial x_n}{\partial z_2} + (x_1 \ x_0) \cdot x_n \cdot \frac{\partial z_n}{\partial z_2} + z_n \cdot (y_1 \ y_0) \cdot \frac{\partial y_n}{\partial z_2} + (y_1 \ y_0) \cdot y_n \cdot \frac{\partial z_n}{\partial z_2}}{z_n^2} \vec{k} \end{aligned} \quad (3.59)$$

Avec

$$\frac{\partial x_n}{\partial x_2} = 0 \quad \frac{\partial y_n}{\partial x_2} = (z_1 \ z_3) \quad \frac{\partial z_n}{\partial x_2} = (y_3 \ y_1);$$

$$\frac{\partial x_n}{\partial y_2} = (z_3 \ z_1) \quad \frac{\partial y_n}{\partial y_2} = 0 \quad \frac{\partial z_n}{\partial y_2} = (x_1 \ x_3);$$

$$\frac{\partial x_n}{\partial z_2} = (y_1 \ y_3) \quad \frac{\partial y_n}{\partial z_2} = (x_3 \ x_1) \quad \frac{\partial z_n}{\partial z_2} = 0;$$

3ième cas, p_0, p_1, p_2 et p_3 forment un quadrilatère et on a ajouté le point p_4 .

On a vu lors du calcul de P , qu'il y avait en tout quatre expressions possibles pour P suivant les égalités entre y_0 et y_1 et entre y_2 et y_3 . En fonction de ces cas on a alors une expression différente pour P à chaque fois.

Cas $y_0 = y_1$ et $y_2 = y_3$, on a alors :

$$\begin{aligned}
 \frac{\partial P}{\partial v_0} &= \frac{(z_0 \ z_1) \frac{\partial y_4}{\partial x_0}}{(y_0 \ y_1)} - \frac{(z_3 \ z_2) \frac{\partial y_4}{\partial x_0}}{(y_3 \ y_2)} \\
 &+ \frac{(z_0 \ z_1) (y_0 \ y_1) \frac{\partial y_4}{\partial y_0}}{(y_0 \ y_1)^2} - \frac{(z_0 \ z_1) (y_4 \ y_1)}{(y_3 \ y_2)} \frac{\partial y_4}{\partial y_0} \\
 &+ \frac{(y_0 \ y_1) + (z_0 \ z_1) \frac{\partial y_4}{\partial z_0}}{(y_0 \ y_1)} - \frac{(z_3 \ z_2) \frac{\partial y_4}{\partial z_0}}{(y_3 \ y_2)}
 \end{aligned} \tag{3.60}$$

Avec en posant $x_4 = \frac{A}{B}$:

$$\begin{aligned}
 \frac{\partial x_4}{\partial x_0} &= \frac{(x_2 (y_3 \ y_1) + x_3 (y_1 \ y_2)) B - A (y_3 \ y_2)}{B^2} \\
 \frac{\partial x_4}{\partial y_0} &= \frac{(x_1 (x_2 \ x_3) B - A (x_2 \ x_3))}{B^2} \\
 \frac{\partial x_4}{\partial z_0} &= 0
 \end{aligned}$$

et en posant également $y_4 = \frac{A}{B}$, on a :

$$\begin{aligned}
 \frac{\partial y_4}{\partial x_0} &= \frac{(y_1 (y_2 \ y_3) B - A (y_2 \ y_3))}{B^2} \\
 \frac{\partial y_4}{\partial y_0} &= \frac{(y_2 (x_3 \ x_1) + y_3 (x_1 \ x_2)) B - A (x_3 \ x_2)}{B^2} \\
 \frac{\partial y_4}{\partial z_0} &= 0
 \end{aligned}$$

Dans les cas suivants les dérivées de x_4 et de y_4 restent les mêmes.

Cas $y_0 = y_1$ et $y_2 = y_3$:

$$\frac{\partial P}{\partial v_0} = \frac{(x_0 \ x_1) (z_0 \ z_1) \frac{\partial x_4}{\partial x_0}}{(x_0 \ x_1)^2} - \frac{(z_3 \ z_2) \frac{\partial y_4}{\partial x_0}}{(y_3 \ y_2)}$$

$$\begin{aligned}
& + \frac{(z_0 \ z_1) \frac{\partial x_4}{\partial y_0}}{(x_0 \ x_1)} \frac{(z_3 \ z_2) \frac{\partial y_4}{\partial y_0} \vec{j}}{(y_3 \ y_2)} \\
& + \frac{(x_4 \ x_1) + (z_0 \ z_1) \frac{\partial x_4}{\partial z_0}}{(x_0 \ x_1)} \frac{(z_3 \ z_2) \frac{\partial y_4}{\partial z_0} \vec{k}}{(y_3 \ y_2)} \quad (3.61)
\end{aligned}$$

Cas $y_0 = y_1$ et $y_2 = y_3$:

$$\begin{aligned}
\frac{\partial P}{\partial v_0} &= \frac{(x_0 \ x_1) (z_0 \ z_1) \frac{\partial x_4}{\partial x_0} (z_0 \ z_1) (x_4 \ x_1)}{(x_0 \ x_1)^2} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial x_0} \vec{i}}{(x_3 \ x_2)} \\
& + \frac{(z_0 \ z_1) \frac{\partial x_4}{\partial y_0}}{(x_0 \ x_1)} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial y_0} \vec{j}}{(x_3 \ x_2)} \\
& + \frac{(x_4 \ x_1) + (z_0 \ z_1) \frac{\partial x_4}{\partial z_0}}{(x_0 \ x_1)} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial z_0} \vec{k}}{(x_3 \ x_2)} \quad (3.62)
\end{aligned}$$

Cas $y_0 = y_1$ et $y_2 = y_3$:

$$\begin{aligned}
\frac{\partial P}{\partial v_0} &= \frac{(z_0 \ z_1) \frac{\partial y_4}{\partial x_0}}{(y_0 \ y_1)} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial x_0} \vec{i}}{(x_3 \ x_2)} \\
& + \frac{(y_0 \ y_1) (z_0 \ z_1) \frac{\partial y_4}{\partial y_0} (z_0 \ z_1) (y_4 \ y_1)}{(y_0 \ y_1)^2} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial y_0} \vec{j}}{(x_3 \ x_2)} \\
& + \frac{(y_4 \ y_1) + (z_0 \ z_1) \frac{\partial y_4}{\partial z_0}}{(y_0 \ y_1)} \frac{(z_3 \ z_2) \frac{\partial x_4}{\partial z_0} \vec{k}}{(x_3 \ x_2)} \quad (3.63)
\end{aligned}$$

Il nous reste enfin à calculer les $\frac{\partial p}{\partial v_i}(i, j)$.

3.5.5.7.2 Calcul des $\frac{\partial p}{\partial v_i}(i, j)$. Il est à nouveau nécessaire de distinguer les trois différents cas.

1er cas, p_0 est au centre du triangle.

En reprenant la formule de :

$$p(i, j) = P \frac{(i \ x_1) (y_1 \ y_2) + (j \ y_1) (x_2 \ x_1)}{x_1 y_2 \ y_1 x_2 + x_0 (y_1 \ y_2) + y_0 (x_2 \ x_1)} \quad (3.64)$$

Que l'on note également sous la forme :

$$p(i, j) = P \frac{B(i, j)}{C} = P A(i, j) \quad (3.65)$$

On en déduit :

$$\frac{\partial p}{\partial v_0}(i, j) = \frac{\partial \vec{P}}{\partial v_0} A(i, j) + P \cdot \frac{B(i, j)}{C^2} ((y_2 \ y_1) \vec{i} + (x_1 \ x_2) \vec{j}) \quad (3.66)$$

2ème cas, p_0 est sur le triangle. On reprend les mêmes notations que précédemment, c'est-à-dire qu'on note p_0 le point du centre et qu'on dérive par rapport à p_2 .

Il faut alors distinguer les sous-cas pour savoir dans quel sous-volume élémentaire on se trouve. A savoir si la droite de projection de coordonnées $x = i$ et $y = j$ passe par le triangle $p_0 p_1 p_2$ ou $p_0 p_2 p_3$ ou $p_0 p_3 p_1$. Dans le triangle $p_0 p_1 p_2$, l'expression de p est alors :

$$p(i, j) = P \cdot \frac{(i \ x_1) \cdot (y_1 \ y_2) + (j \ y_1) \cdot (x_2 \ x_1)}{(x_1 \ x_0) \cdot (y_2 \ y_0) \ (y_1 \ y_0) \cdot (x_2 \ x_0)} \quad (3.67)$$

On a alors :

$$\begin{aligned} \frac{\partial p}{\partial v_2}(i, j) &= \frac{\partial \vec{P}}{\partial v_2} \cdot \frac{A(i, j)}{B} + P \cdot \frac{(B \cdot (j \ y_1) \ A(i, j) \cdot (y_0 \ y_1)) \vec{i}}{B^2} \\ &\quad + P \cdot \frac{(B \cdot (x_1 \ i) \ A(i, j) \cdot (x_1 \ x_0)) \vec{j}}{B^2} \end{aligned} \quad (3.68)$$

Dans le triangle $p_0p_2p_3$:

$$p(i, j) = P \cdot \frac{(i \ x_2) \cdot (y_2 \ y_3) + (j \ y_2) \cdot (x_3 \ x_2)}{(x_2 \ x_0) \cdot (y_3 \ y_0) \ (y_2 \ y_0) \cdot (x_3 \ x_0)} \quad (3.69)$$

On a alors :

$$\begin{aligned} \frac{\partial p}{\partial v_2}(i, j) &= \frac{\partial \vec{P}}{\partial v_2} \cdot \frac{A(i, j)}{B} + P \cdot \frac{(B \cdot (y_3 \ j) \ A(i, j) \cdot (y_3 \ y_0)) \vec{i}}{B^2} \\ &\quad + P \cdot \frac{(B \cdot (i \ x_3) \ A(i, j) \cdot (x_0 \ x_3)) \vec{j}}{B^2} \end{aligned} \quad (3.70)$$

Et enfin dans $p_0p_3p_1$:

$$p(i, j) = P \cdot \frac{(i \ x_3) \cdot (y_3 \ y_1) + (j \ y_3) \cdot (x_1 \ x_3)}{(x_3 \ x_0) \cdot (y_1 \ y_0) \ (y_3 \ y_0) \cdot (x_1 \ x_0)} \quad (3.71)$$

On a alors :

$$\frac{\partial p}{\partial v_2}(i, j) = \frac{\partial \vec{P}}{\partial v_2} \cdot \frac{A(i, j)}{B} \quad (3.72)$$

3ième cas, $p_0p_1p_2p_3$ forment un quadrilatère et on a ajouté un point p_4 .

On doit alors distinguer 4 sous cas en fonction du triangle $p_4p_2p_0$ ou $p_4p_0p_3$ ou $p_4p_3p_1$ ou $p_4p_1p_2$ par lequel la droite de projection passe.

Dans le triangle $p_4p_2p_0$:

$$p(i, j) = P \cdot \frac{(i \ x_2) \cdot (y_2 \ y_0) + (j \ y_2) \cdot (x_0 \ x_2)}{(x_2 \ x_4) \cdot (y_0 \ y_4) \ (y_2 \ y_4) \cdot (x_0 \ x_4)} = P \cdot \frac{A(i, j)}{B} \quad (3.73)$$

et donc :

$$\frac{\partial p}{\partial v_0}(i, j) = \frac{\partial \vec{P}}{\partial v_0} \cdot \frac{A(i, j)}{B} + \frac{P}{B^2} \cdot \left(B \cdot \frac{\partial A(i, j)}{\partial x_0} - A(i, j) \cdot \frac{\partial B}{\partial x_0} \right) \vec{i} + \frac{P}{B^2} \cdot \left(B \cdot \frac{\partial A(i, j)}{\partial y_0} - A(i, j) \cdot \frac{\partial B}{\partial y_0} \right) \vec{j} \quad (3.74)$$

Avec :

$$\frac{\partial A(i, j)}{\partial x_0} = j - y_2$$

$$\frac{\partial A(i, j)}{\partial y_0} = x_2 - i$$

$$\frac{\partial B}{\partial x_0} = y_4 - y_2 + (y_2 - y_0) \cdot \frac{\partial x_4}{\partial x_0} + (x_0 - x_2) \cdot \frac{\partial y_4}{\partial x_0}$$

$$\frac{\partial B}{\partial y_0} = x_2 - x_4 + (y_2 - y_0) \cdot \frac{\partial x_4}{\partial y_0} + (x_0 - x_2) \cdot \frac{\partial y_4}{\partial y_0}$$

Dans le triangle $p_4p_0p_3$,

$$p(i, j) = P \cdot \frac{(i - x_0) \cdot (y_0 - y_3) + (j - y_0) \cdot (x_3 - x_0)}{(x_0 - x_4) \cdot (y_3 - y_4) - (y_0 - y_4) \cdot (x_3 - x_4)} = P \cdot \frac{A(i, j)}{B} \quad (3.75)$$

On a la même formule que précédemment et :

$$\frac{\partial A(i, j)}{\partial x_0} = y_3 - y_0 - j + y_0$$

$$\frac{\partial A(i, j)}{\partial y_0} = i - x_0 - x_3 + x_0$$

$$\frac{\partial B}{\partial x_0} = y_3 - y_4 + (y_0 - y_3) \cdot \frac{\partial x_4}{\partial x_0} + (x_3 - x_0) \cdot \frac{\partial y_4}{\partial x_0}$$

$$\frac{\partial B}{\partial y_0} = x_4 - x_3 + (y_0 - y_3) \cdot \frac{\partial x_4}{\partial y_0} + (x_3 - x_0) \cdot \frac{\partial y_4}{\partial y_0}$$

Dans le triangle $p_4p_3p_1$:

$$p(i, j) = P \cdot \frac{(i - x_3) \cdot (y_3 - y_1) + (j - y_3) \cdot (x_1 - x_3)}{(x_3 - x_4) \cdot (y_1 - y_4) - (y_3 - y_4) \cdot (x_1 - x_4)} = P \cdot \frac{A(i, j)}{B} \quad (3.76)$$

et :

$$\frac{\partial A(i, j)}{\partial x_0} = 0$$

$$\frac{\partial A(i, j)}{\partial y_0} = 0$$

$$\frac{\partial B}{\partial x_0} = (y_3 \quad y_1) \cdot \frac{\partial x_4}{\partial x_0} + (x_1 \quad x_3) \cdot \frac{\partial y_4}{\partial x_0}$$

$$\frac{\partial B}{\partial y_0} = (y_3 \quad y_1) \cdot \frac{\partial x_4}{\partial y_0} + (x_1 \quad x_3) \cdot \frac{\partial y_4}{\partial y_0}$$

Et enfin dans le triangle $p_4 p_1 p_2$:

$$p(i, j) = P \cdot \frac{(i \quad x_1) \cdot (y_1 \quad y_2) + (j \quad y_1) \cdot (x_2 \quad x_1)}{(x_1 \quad x_4) \cdot (y_2 \quad y_4) \quad (y_1 \quad y_4) \cdot (x_2 \quad x_4)} = P \cdot \frac{A(i, j)}{B} \quad (3.77)$$

et :

$$\frac{\partial A(i, j)}{\partial x_0} = 0$$

$$\frac{\partial A(i, j)}{\partial y_0} = 0$$

$$\frac{\partial B}{\partial x_0} = (y_1 \quad y_2) \cdot \frac{\partial x_4}{\partial x_0} + (x_2 \quad x_1) \cdot \frac{\partial y_4}{\partial x_0}$$

$$\frac{\partial B}{\partial y_0} = (y_1 \quad y_2) \cdot \frac{\partial x_4}{\partial y_0} + (x_2 \quad x_1) \cdot \frac{\partial y_4}{\partial y_0}$$

Tous les éléments nécessaires au calcul du gradient ont donc été déterminés, on peut alors résumer l'algorithme effectué pour ce calcul du gradient.

3.5.6 Résumé de l'algorithme

Pour tous les points v_l

Pour tous les repères orientés suivant θ

Calcul du point supplémentaire e et de signe

(signe est le signe de $\overrightarrow{bar{y}} \cdot \vec{n}_l$ et détermine si le volume est extérieur ou intérieur)

Pour tous les voisins de v_l

(on prend à chaque fois deux voisins, le point v_l lui-même et le point e de façon à avoir un volume élémentaire)

Calcul de P et de $\frac{\partial P}{\partial v_l}$

Pour tous les i et j

Calcul de $p(i, j)$ et $\frac{\partial p}{\partial v_l}(i, j)$ au travers du volume considéré

$$\vec{k}_{k,i,j} = \text{signe} \cdot 2 \cdot \frac{\partial p}{\partial v_l}(i, j) \cdot (p_{k,i,j} \quad d_{k,i,j})$$

On exprime $\vec{grad}_{k,i,j}$ dans le repère de référence

$$\vec{grad}_l = \vec{grad}_l + \vec{grad}_{k,i,j}$$

CHAPITRE 4

RÉSULTATS EXPÉRIMENTAUX.

Nous allons à présent présenter quelques résultats afin de pouvoir illustrer les capacités et les limites de cette méthode. Nous verrons en particulier les problèmes qui apparaissent lorsque les formes présentent trop d'occlusions.

4.1 Présentation des conditions des expériences

Rappelons tout d'abord les conditions avec lesquelles nous faisons tourner le programme. La procédure se fait en deux étapes. La première étape, qui n'est pas propre à notre algorithme, consiste à récupérer des cartes de profondeurs comme nous l'avons déjà mentionné dans les parties précédentes. Pour cela, nous utiliserons des objets synthétiques et nous prendrons les mêmes mesures que celles que l'on peut obtenir par stéréovision, c'est-à-dire la profondeur des points de la scène vue à partir d'un point précis. La forme ou l'objet est réalisé avec un logiciel de dessin en trois dimensions. Le premier sous programme va donc lire le fichier du dessin, puis va ensuite l'analyser pour en déduire les données recherchées. Afin de pouvoir lire ce fichier, nous sauvegardons celui-ci sous le format .ase (format très classique et basique utilisé par les logiciels de dessin en 3 dimensions) qui est très intuitif et qui permet de récupérer facilement la liste des points de la figure ainsi que les arêtes. Après cette lecture on affiche alors à l'écran la forme à l'aide de la librairie OpenGL puis en utilisant une de ses fonctions, on récupère la profondeur des pixels affichés. On fait ensuite pivoter cette figure de 180 degrés de façon à afficher l'arrière de la forme, puis en récupérant à nouveau la profondeur des pixels affichés et en les



FIG. 4.1 – Une forme représentée sous OpenGL vue de profil, puis vue après une rotation de 180 degrés. La correspondance des points de ces deux vues avec leurs profondeurs permet de constituer une carte de profondeurs.

corrélant à l'autre face, on en déduit la carte de profondeurs. Lors de l'appariement entre les pixels de la face avant et arrière, il faudra veiller aux inversions produites lors de la rotation de 180 degrés. Si celle-ci a été faite autour de l'axe vertical par exemple, les points qui étaient à droite sur la face avant correspondent aux points qui se trouvent à gauche sur la face arrière et réciproquement. Rajoutons enfin qu'au lancement du programme, on entre les directions d'observation que l'on veut relever. On précise pour cela à chaque fois la rotation autour de l'axe des X et celle autour de l'axe des Y . Dans la plupart des cas, on choisit la vue de face, celle du dessus et celle de côté, soit encore entrant les angles suivants : $\theta_{x_0} = 0$, $\theta_{y_0} = 0$, $\theta_{x_1} = 90$, $\theta_{y_1} = 0$ et $\theta_{x_2} = 0$, $\theta_{y_2} = 90$. L'ensemble des données calculées est alors écrit dans un fichier qui sera ensuite lu par notre programme principal.

Pour celui-ci nous entrons également plusieurs paramètres. Comme nous l'avons

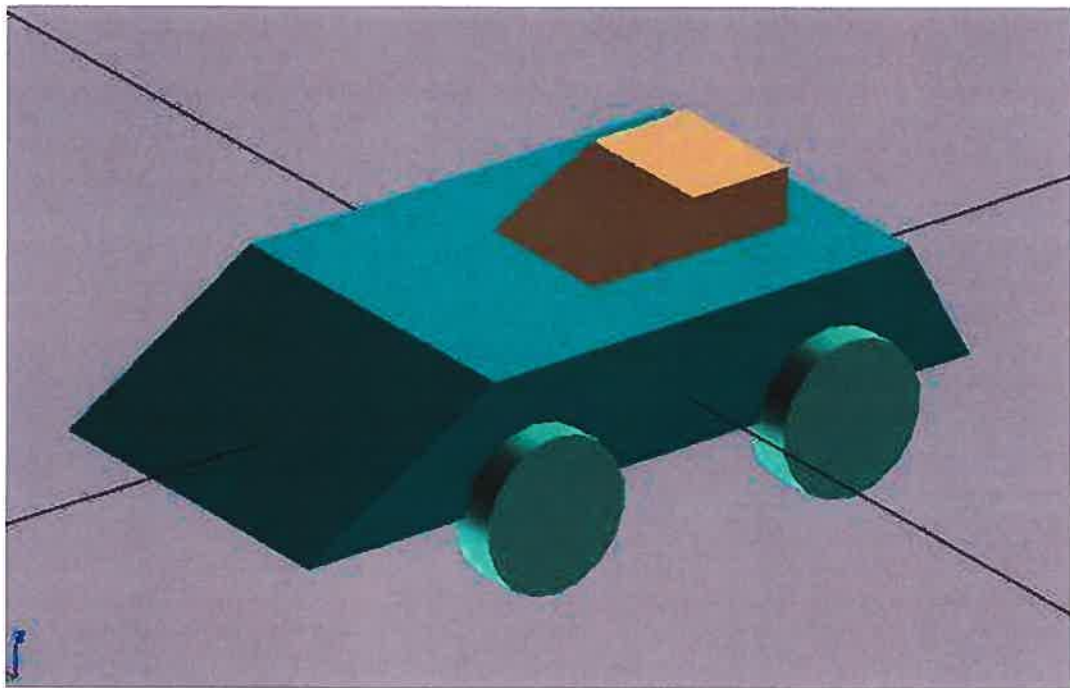
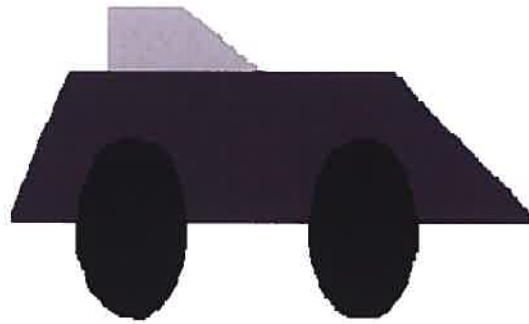


FIG. 4.2 – Première image : Schématisation d'une carte de profondeurs. Les parties les plus noires sont les plus épaisses et inversement pour les parties les plus blanches. Deuxième image : voiture d'origine en vue oblique.

expliqué, l'idée principale de l'algorithme est de diminuer une fonction d'énergie qui correspond à la différence entre les données de départ et celles que l'on obtient avec la forme que l'on modèle. Deux critères d'arrêts peuvent apparaître intuitifs. Le premier serait d'arrêter le programme lorsque la fonction d'énergie descend en dessous d'un certain seuil. Cependant, celle-ci ne tend pas vers 0 donc on ne sait pas vers quelle valeur celle-ci va se stabiliser. En effet, d'une part les points sont déplacés avec un certain pas, donc lorsque l'on arrive à un minimum, à chaque déplacement, on oscille autour de ce minimum sans l'atteindre. Et d'autre part, comme nous l'avons expliqué dans la partie précédente, certaines données se contredisent lorsqu'il y a des concavités et on ne peut aboutir à une solution parfaite. Un autre critère d'arrêt pourrait être de s'arrêter dès que la fonction d'énergie prend une valeur supérieure à celle de l'itération précédente, mais ce phénomène se produit parfois bien avant la stabilisation. Ceci s'explique pour les mêmes raisons que précédemment, un pas trop grand aboutit parfois à un état moins bon que le précédent. On choisit donc finalement d'entrer en paramètre au lancement, le nombre d'itérations à effectuer.

Un autre critère que l'on entre, est le coefficient que l'on place devant le gradient. Chaque point est déplacé suivant l'opposé du vecteur gradient auquel on applique un coefficient évitant des débordements dans les déplacements. Par expérience, on fera varier celui-ci entre 0.01 et 0.001. On comprendra par ailleurs que plus le pas est petit, plus on choisira un nombre d'itérations important.

Vient ensuite le choix entre deux méthodes. La première consiste à partir d'un état de notre forme, de calculer les gradients par rapport à tous les points et déplacer tous les points lors d'une seule et même itération. La seconde méthode, plus

lente, consiste à déplacer les points un par un. Ainsi, chaque gradient calculé tient compte des points précédemment déplacés. Cette seconde méthode étant bien plus longue, et l'expérience nous montrant que les résultats obtenus n'apparaissent pas meilleurs, on choisira donc presque tout le temps la première.

Enfin, on détermine le nombre de points de la forme que l'on va modéliser. Comme nous l'avons vu précédemment, la construction de notre sphère initiale se fait par divisions successives de facettes triangulaires, et le nombre de points que l'on peut envisager se fait donc par paliers. On prendra en général entre 34 et 130 points.

Pour la plupart des résultats présentés ci-après, on indiquera les angles de vue pris sur l'objet, le nombre d'itérations ainsi que le pas (ce dernier détail n'a d'intérêt que pour l'exécution du programme). Pour les vues utilisées, on précise à chaque fois l'angle autour de l'axe x que l'on note θ_x puis celui autour de l'axe y que l'on note θ_y . La vue $\theta_x = 0$, $\theta_y = 0$ étant toujours celle par défaut, on ne la précise pas. Ainsi, si l'on note "vue 90 0 0 90 itérations 200 pas 0.01", cela signifie que l'on a pris trois angles de vue pour nos mesures. La première vue par défaut est effectuée avec $\theta_x = 0$ et $\theta_y = 0$, la deuxième avec $\theta_x = 90$ et $\theta_y = 0$ et la troisième avec $\theta_x = 0$ et $\theta_y = 90$. Le nombre d'itérations effectuées est de 200 et le pas était de 0.01.

Enfin, nous avons utilisé un logiciel annexe qui permet de recalculer deux formes similaires et de faire des calculs sur les différences entre ces deux formes. On fera donc apparaître des images de superposition ainsi que des diagrammes indiquant la répartition des erreurs au niveau de la surface des deux formes. Il faut noter que les images sont contenues dans des fenêtres carrées de cent millimètres de côté. Si on enlève les marges, les figures étudiées tiennent donc approximativement dans

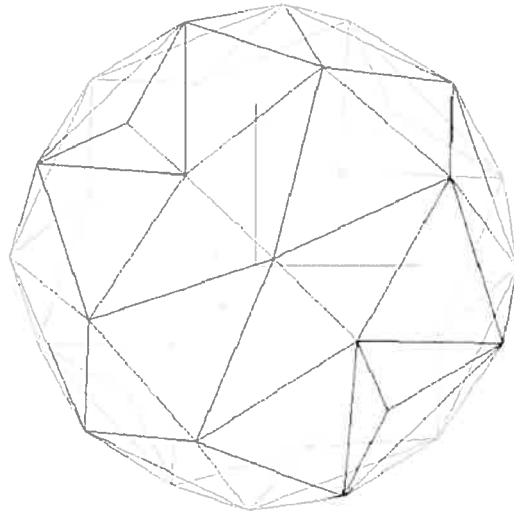


FIG. 4.3 – sphère vue 90 0 itérations 300 pas 0.01

une fenêtre carrée de quatre vingt millimètres de côté. Le terme "rms" correspond à la valeur efficace ("root mean square").

4.2 Tests basiques

La première forme de base que l'on teste est une sphère. Deux vues de face et de côté suffisent pour obtenir un bon résultat (voir figure 4.3). On observe sur le diagramme 4.4 que l'erreur est très faible.

On fait ensuite le test suivant sur un cube. Là aussi, pas de problème majeur, le résultat est correct, 2 vues suffisent (voir figure 4.5).

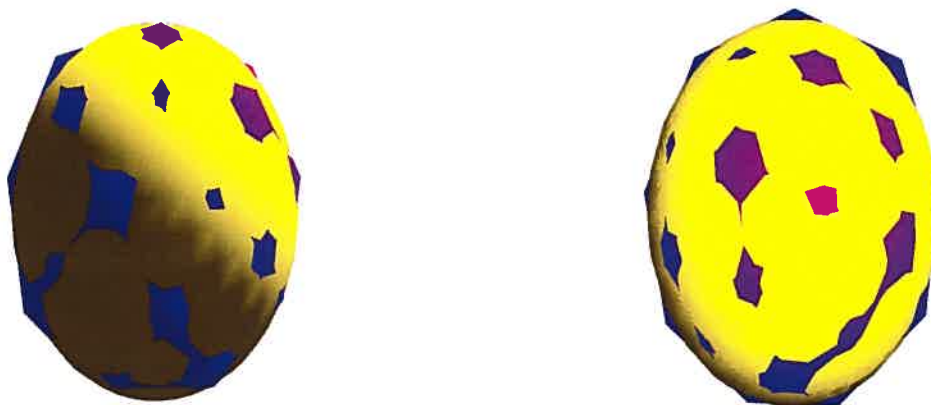
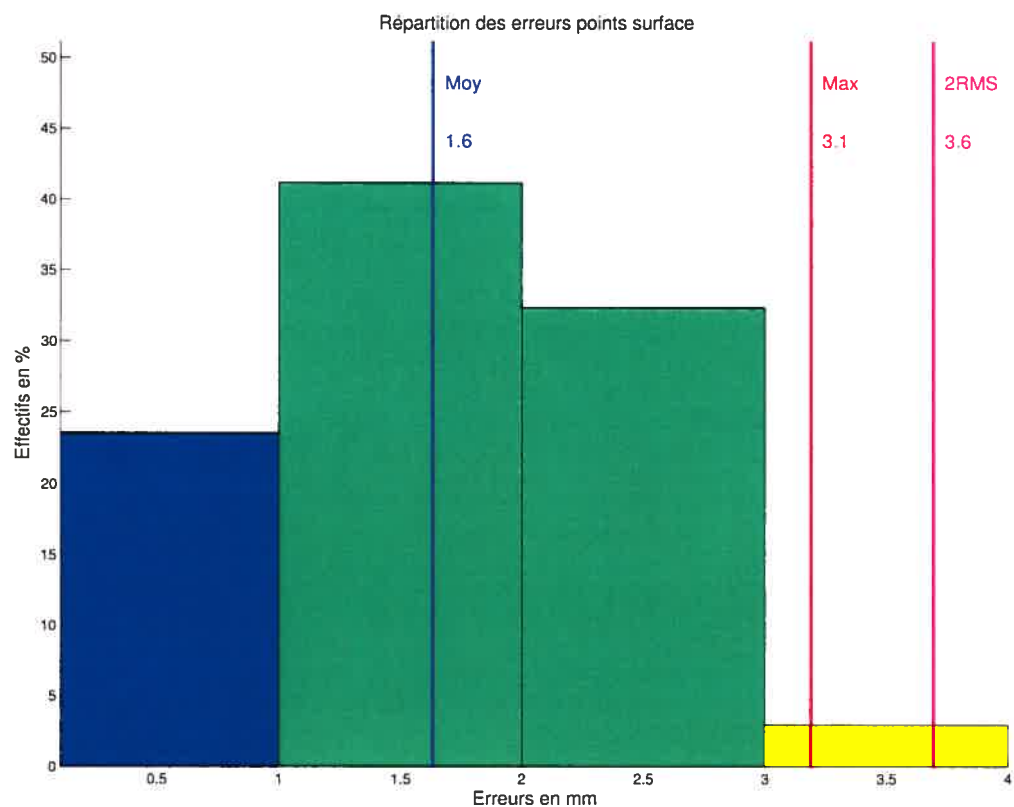


FIG. 4.4 – Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue. Nous rappelons que cette figure, ainsi que les suivantes sont contenues dans des fenêtres carrées de cent millimètres de côté avec une marge d'environ cinq à dix millimètres sur les bords.

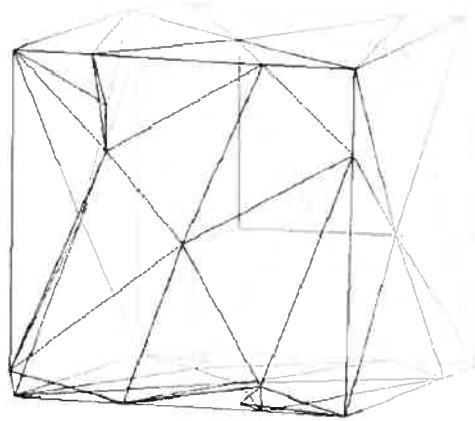


FIG. 4.5 – Cube, vue 90 0 itérations 300 pas 0.01

4.3 Tests avec des formes plus évoluées

On fait ensuite un test avec un objet en forme de lampe. En choisissant les vues dans des bonnes directions, en particulier celles où il n'y a pas d'occlusion, on arrive encore une fois à obtenir la forme de départ. On remarque au niveau de la répartition des erreurs quelques points isolés qui correspondent à quelques surfaces fines qui s'écartent du pied de la lampe mais qui, par leur faible épaisseur, ne représentent pas d'erreurs importantes au niveau des cartes de projection. On montre ici les résultats intermédiaires sur les figures 4.6, 4.7 et 4.8, ainsi que la répartition des erreurs et quelques images de superposition : figure 4.10.

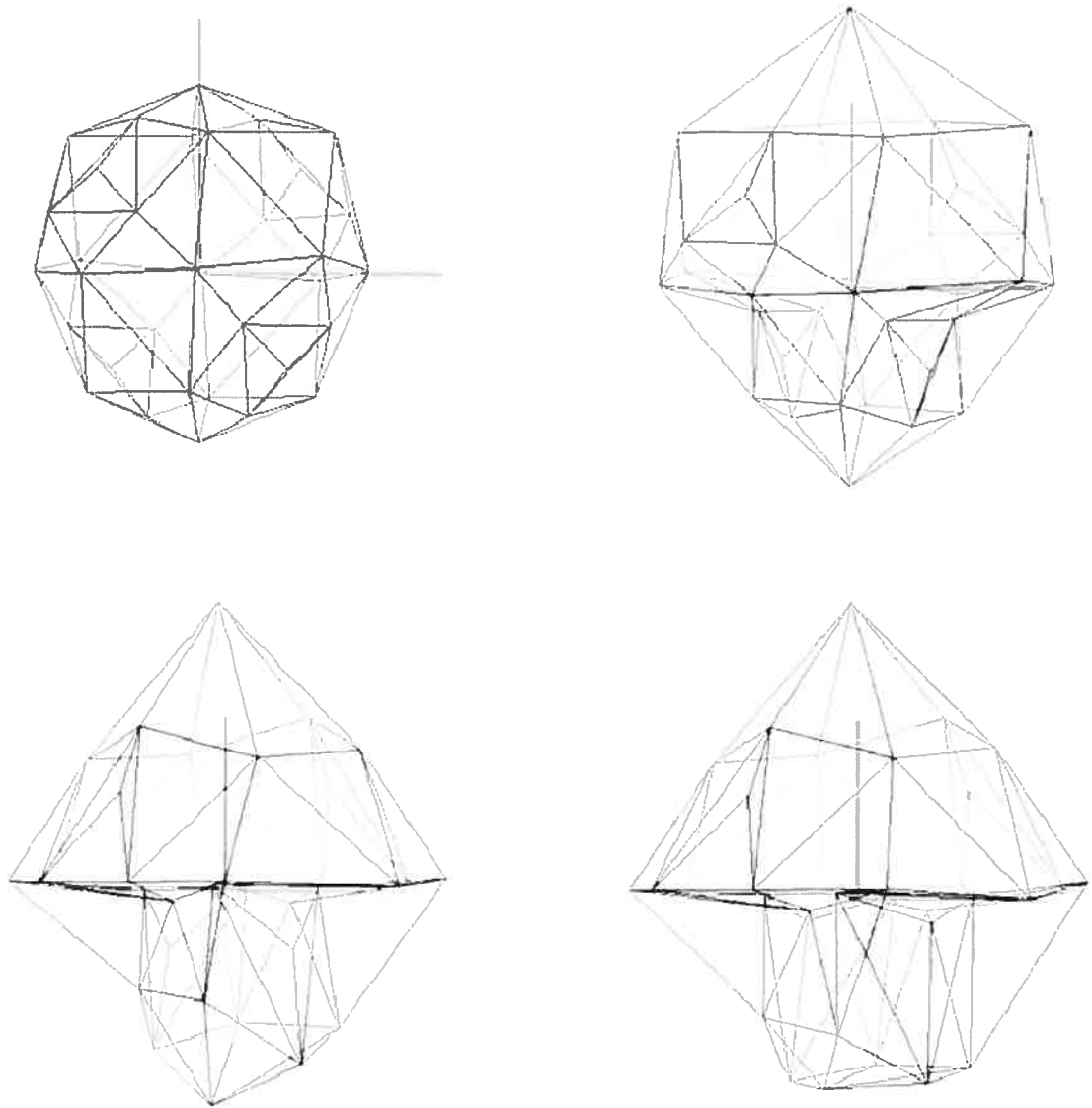


FIG. 4.6 – 10 itérations, 50 itérations, 100 itérations et 150 itérations

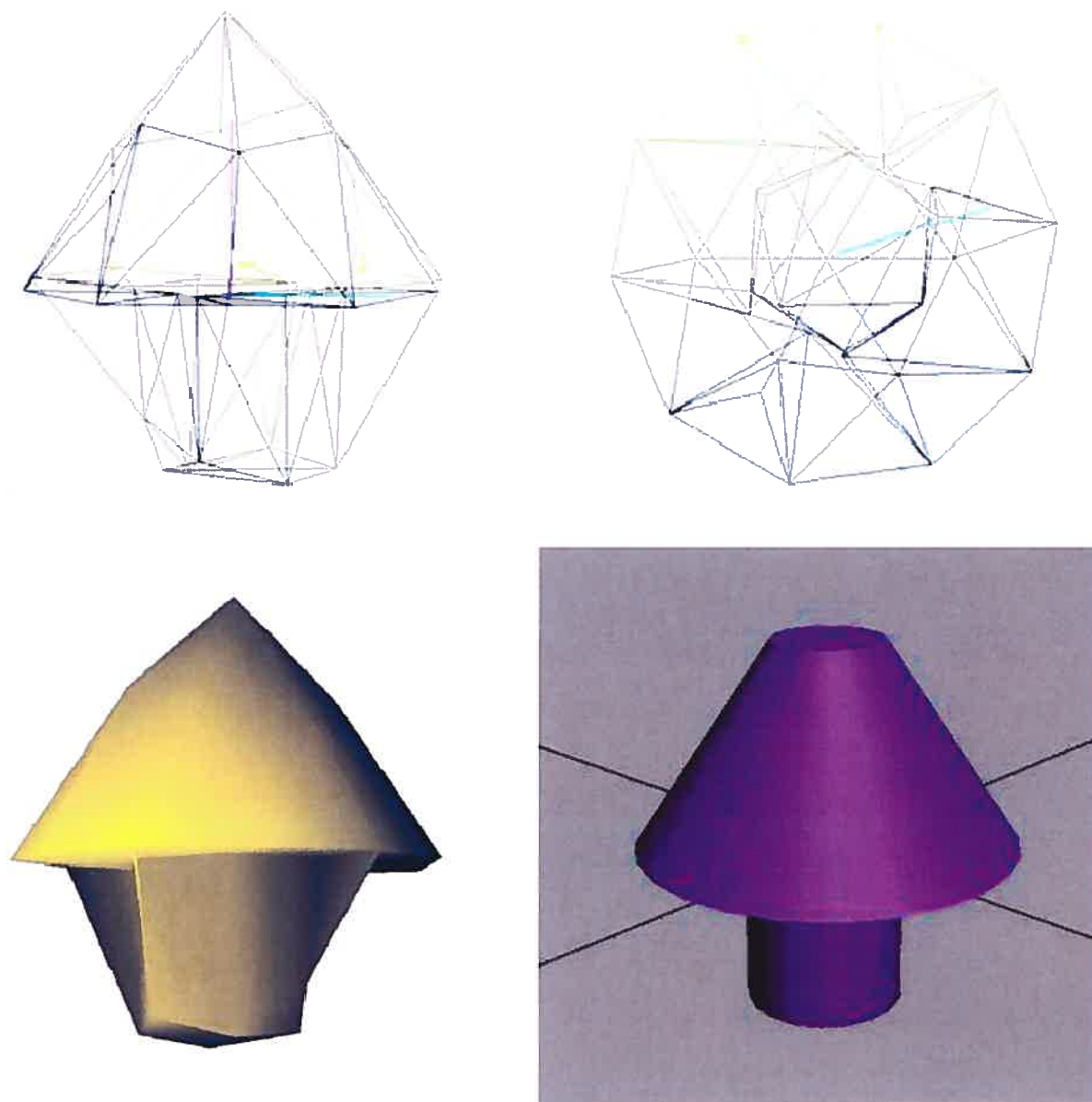


FIG. 4.7 – Lampe, vues 90 0 0 90 itérations 200 pas 0.01

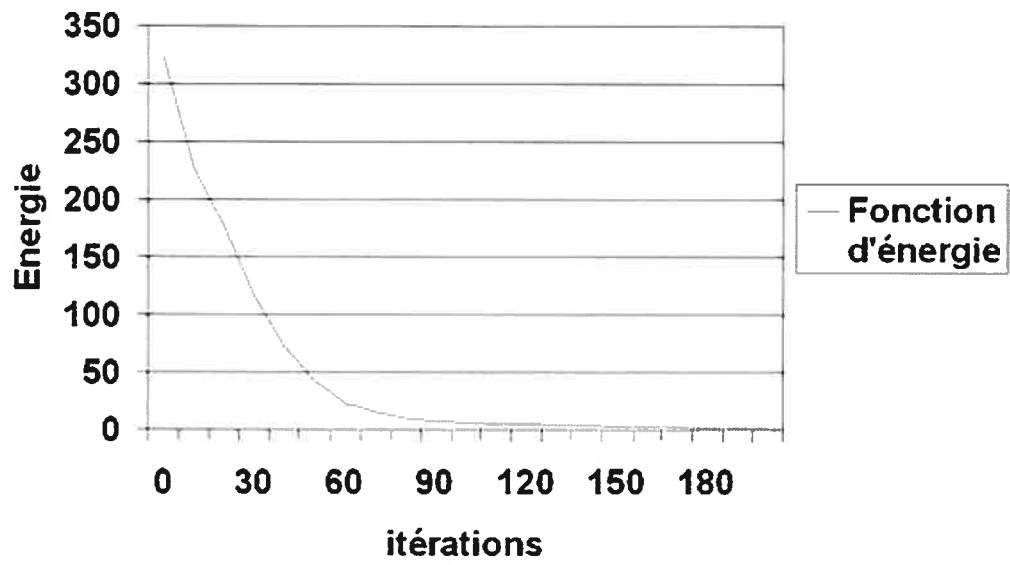


FIG. 4.8 – Représentation de la courbe de la fonction d'énergie calculée au cours des itérations.

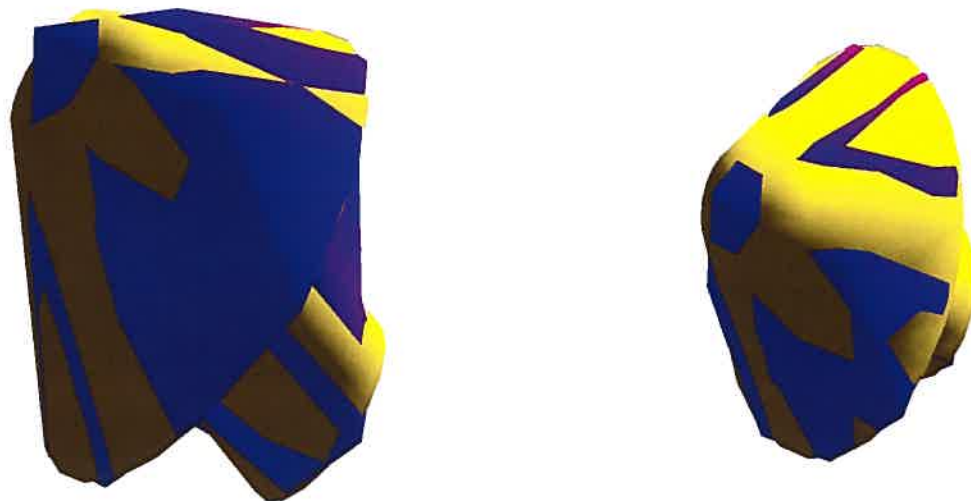
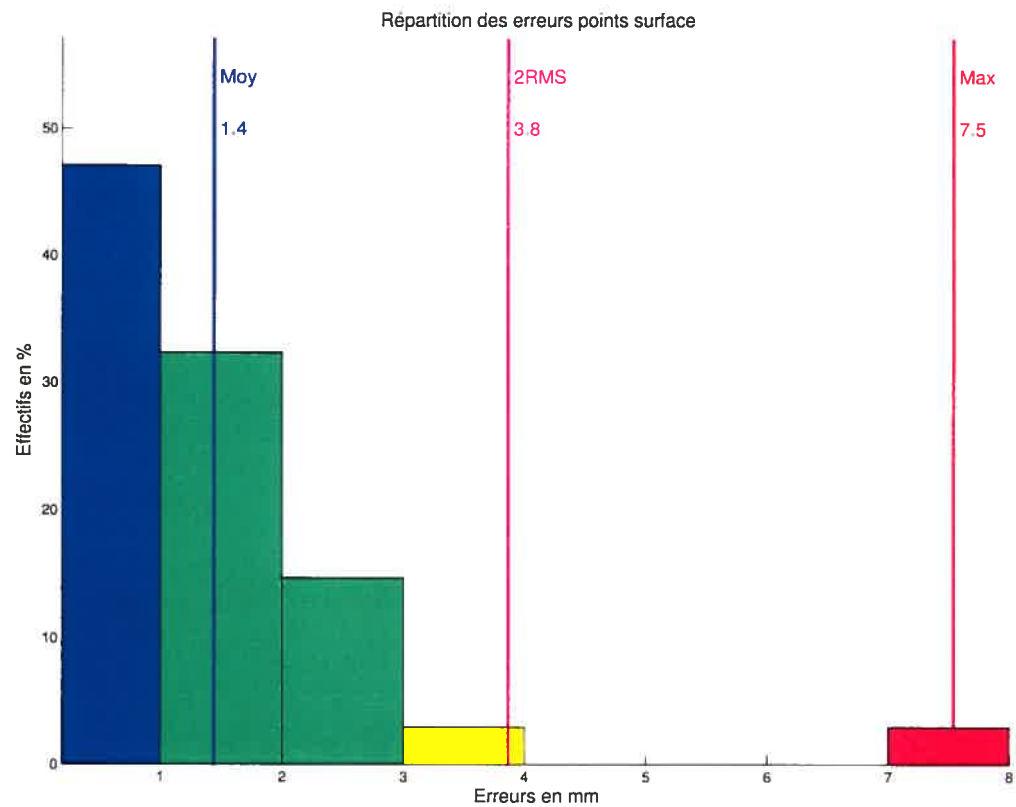


FIG. 4.9 – Diagramme de répartition d'erreurs. Superposition de l'image de référence et de l'image obtenue selon deux points de vue.

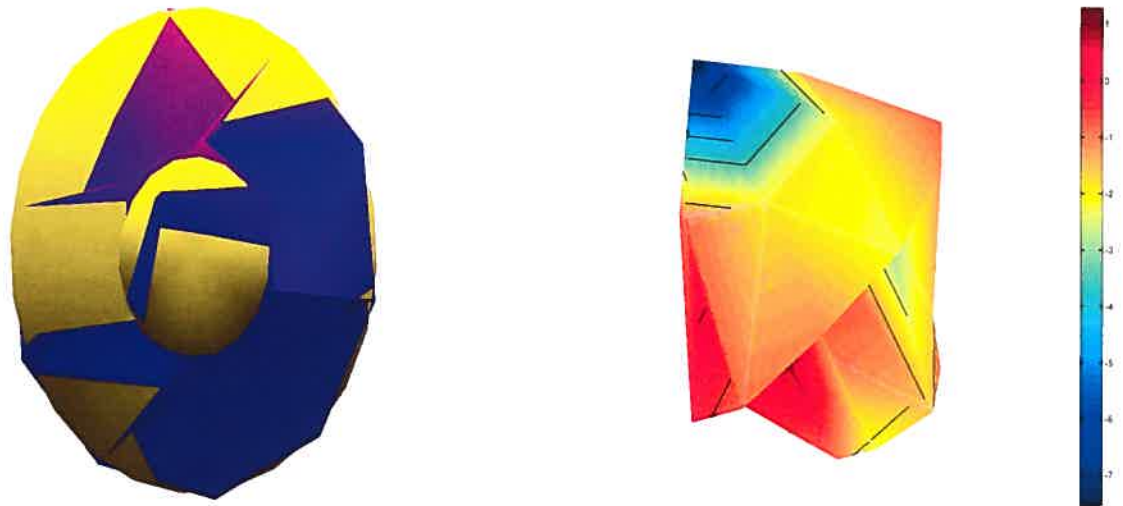


FIG. 4.10 – Superposition de l'image de référence et de l'image obtenue. Cartographie de l'erreur : les couleurs extrêmes rouge ou bleu foncée indiquent de fortes erreurs, celles entre le vert et le jaune indiquent des erreurs plus faibles.

On observe ensuite un résultat obtenu avec une forme simple un cône avec l'extrémité inférieure en forme de demie sphère (voir figures 4.11 et 4.23). Le résultat est très satisfaisant.

Nous faisons ensuite un test avec une forme d'avantage non convexe. En choisissant des vues autres que celles de côté, on retrouve la forme visée à la figure 4.13. Au niveau des erreurs, on voit que celles-ci sont réparties. Les bords extérieurs sont assez respectés mais les deux axes engendrent des erreurs plus importantes (voir figure 4.14).

4.4 Mise en évidence de l'importance du choix des vues et du problème des occlusions

Pour bien illustrer le problème des occlusions, nous faisons un test avec une vue de face et une vue de côté. Celle de côté n'enregistre donc pas qu'il y a un vide entre les deux pieds de « l'arche ». Il y a donc confusion entre les deux vues. La forme obtenue n'est pas du tout celle recherchée (voir figure 4.15). Aucun creux ne se forme au centre.

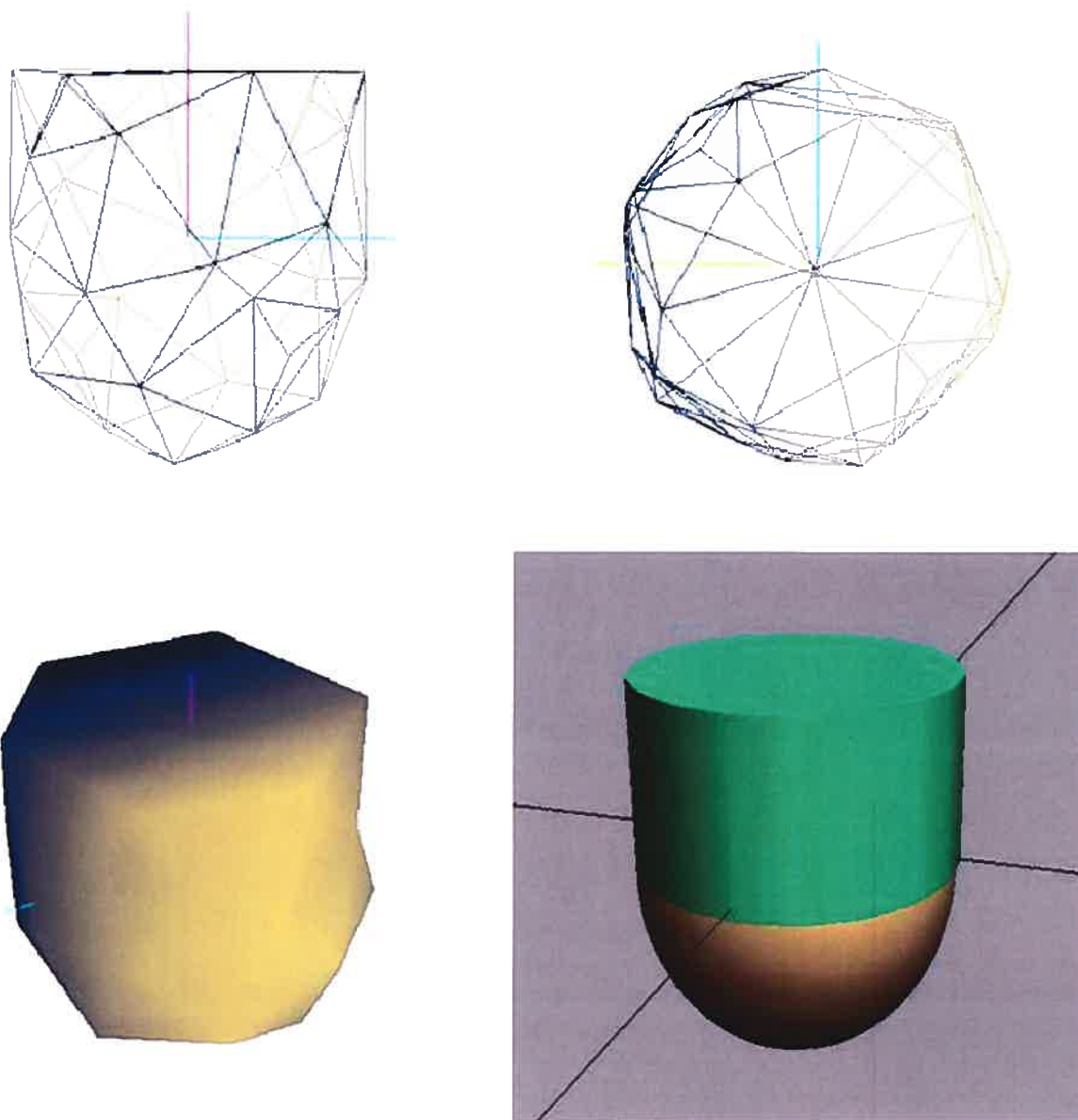


FIG. 4.11 – Cylindre-sphère vues 90 0 méthode 0 itérations 200 pas 0.01

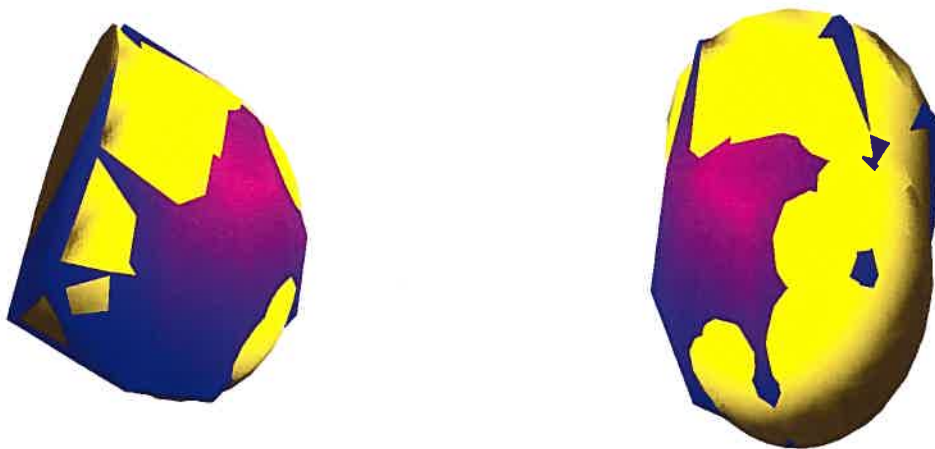
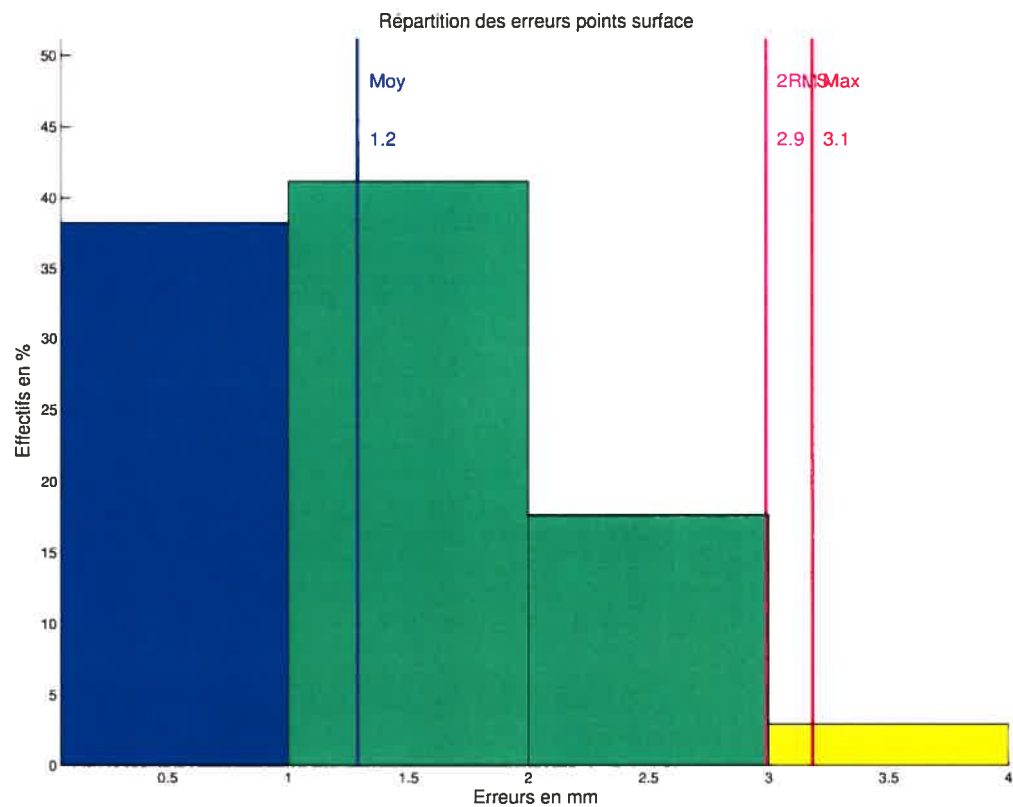


FIG. 4.12 – Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.

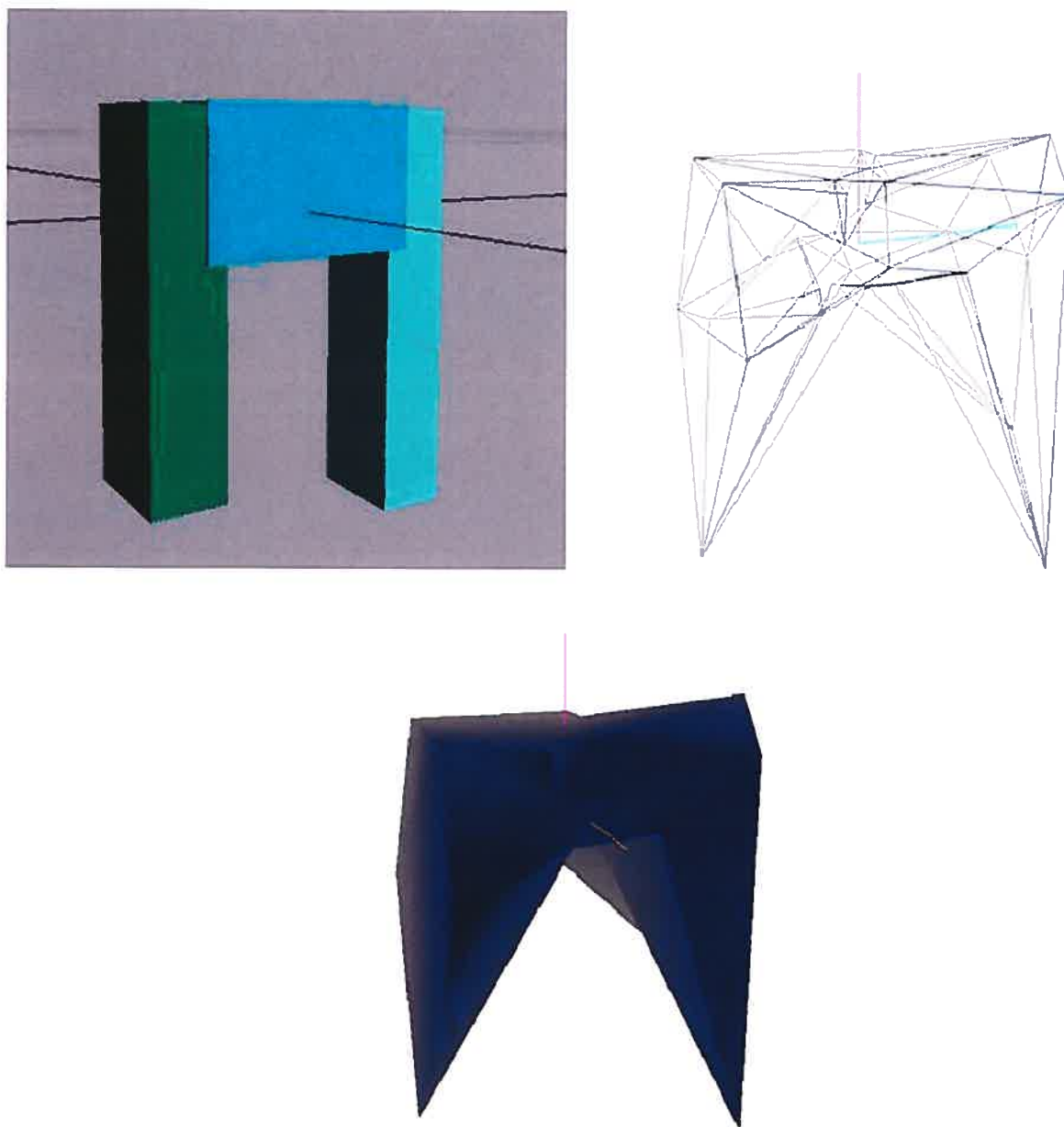


FIG. 4.13 – Arche vues 90 0 0 90 méthode 0 itérations 200 pas 0.01

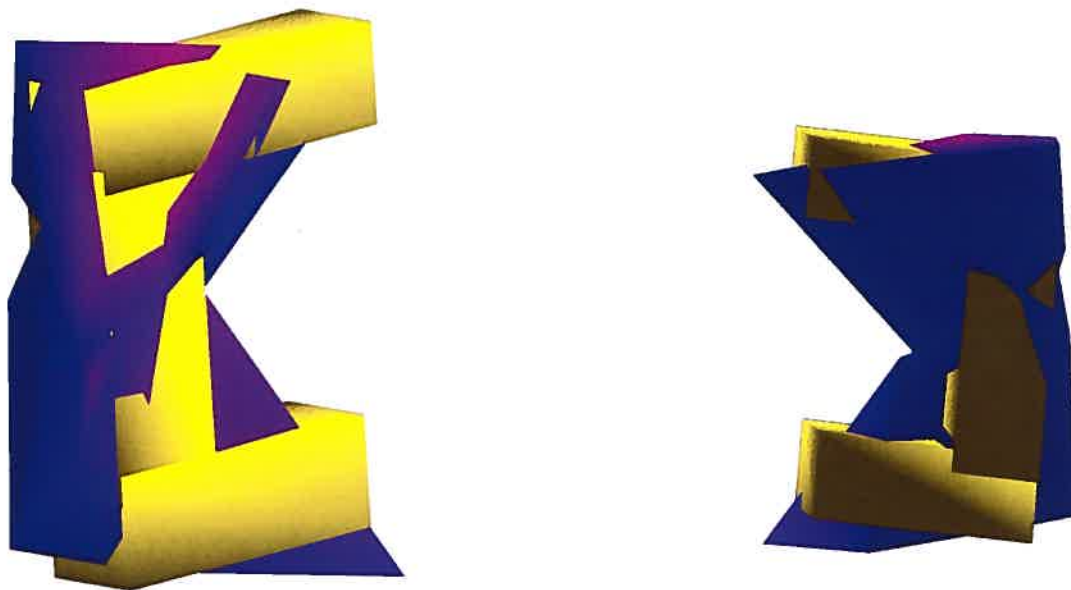
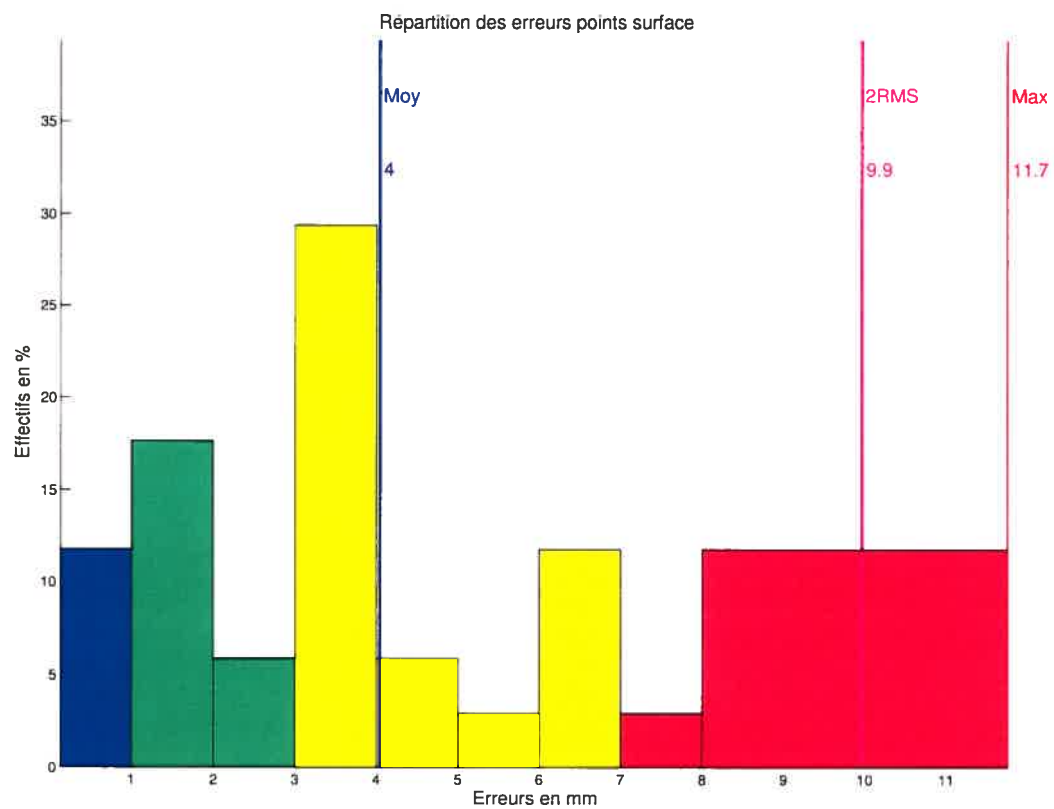


FIG. 4.14 – Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.

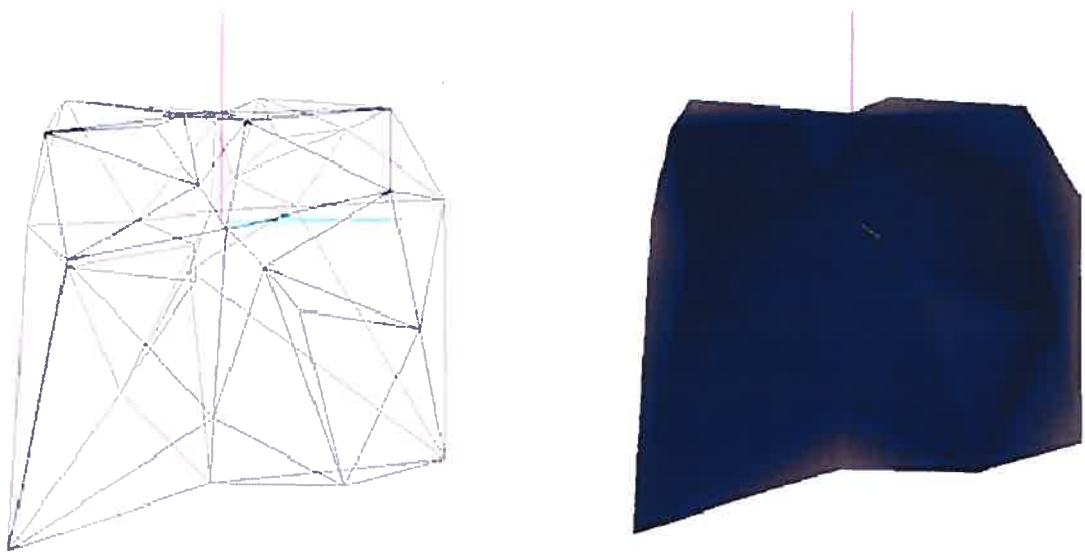


FIG. 4.15 – Arche vues 90 0 0 90 méthode 0 itérations 200 pas 0.01

Nous venons donc de montrer l'importance du choix des vues et, par ce fait, on montre également qu'augmenter le nombre de vues n'améliore pas nécessairement le résultat. Illustrons à nouveau cette idée en augmentant progressivement le nombre de vues pour un même objet. Nous reprenons à nouveau la "lampe" comme objet d'étude qui n'est ni trop simple ni trop complexe. On précise pour ces résultats les vues prises ainsi que la valeur de la fonction d'énergie et nous donnons une vue de face et une vue de dessus à chaque fois.

Nous observons qu'avec une vue, la fonction d'énergie descend très bas. Il n'y a aucune "gêne" ou déplacement opposé ou du moins différent entraîné par plusieurs prises de vue. En revanche, si la forme vue de face est bien respectée, il n'y a aucune contrainte sur les autres vues et on s'aperçoit, en particulier avec la vue de haut, que la forme n'est pas circulaire. Dans le deuxième test, nous ajoutons aux données une vue de dessus (vue : 90 0), cette fois-ci la fonction d'énergie est plus élevée (1.60) mais la forme est bien mieux respectée et est à présent circulaire (voir figure 4.16). Remarquons que cela n'a pas beaucoup de sens de comparer l'énergie d'un résultat pris avec un nombre de vues différent mais le plus cohérent pour cette comparaison serait de diviser cette énergie par le nombre de vues. En effet, la fonction d'énergie est une somme de différences que l'on effectue pour chaque vue. Donc pour n vues, il y a n fois plus de termes dans cette somme. Ici le nombre à retenir pourrait être $\frac{1.60}{2} = 0.8$.

En passant à trois vues, le résultat ne change pas vraiment et le terme de comparaison de la fonction d'énergie $\frac{2.05}{3} = 0.68$ ne varie pas beaucoup. La nouvelle vue de côté n'est pas contradictoire et ne fait pas apparaître de problème d'occlusion. En revanche, avec l'ajout d'une quatrième vue oblique, le terme de comparaison grimpe : $\frac{8.1}{4} = 2.03$, car les problèmes d'occlusion et d'incohérence entre les vues apparaissent (voir figure 4.17).

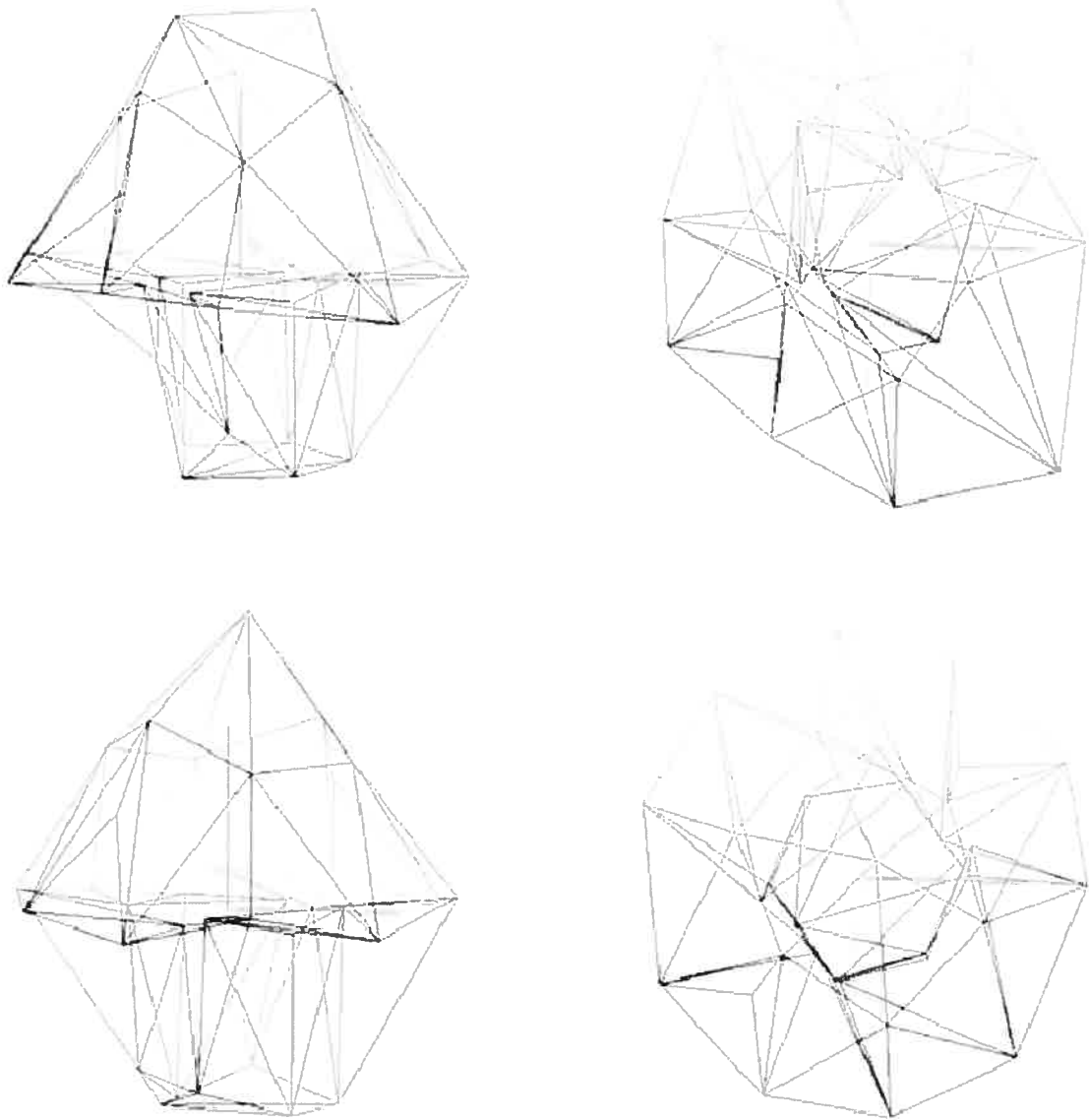


FIG. 4.16 – Images du haut : vue 0 0, énergie : 0.25, images du bas : vue 0 0 90 0, énergie : 1.60

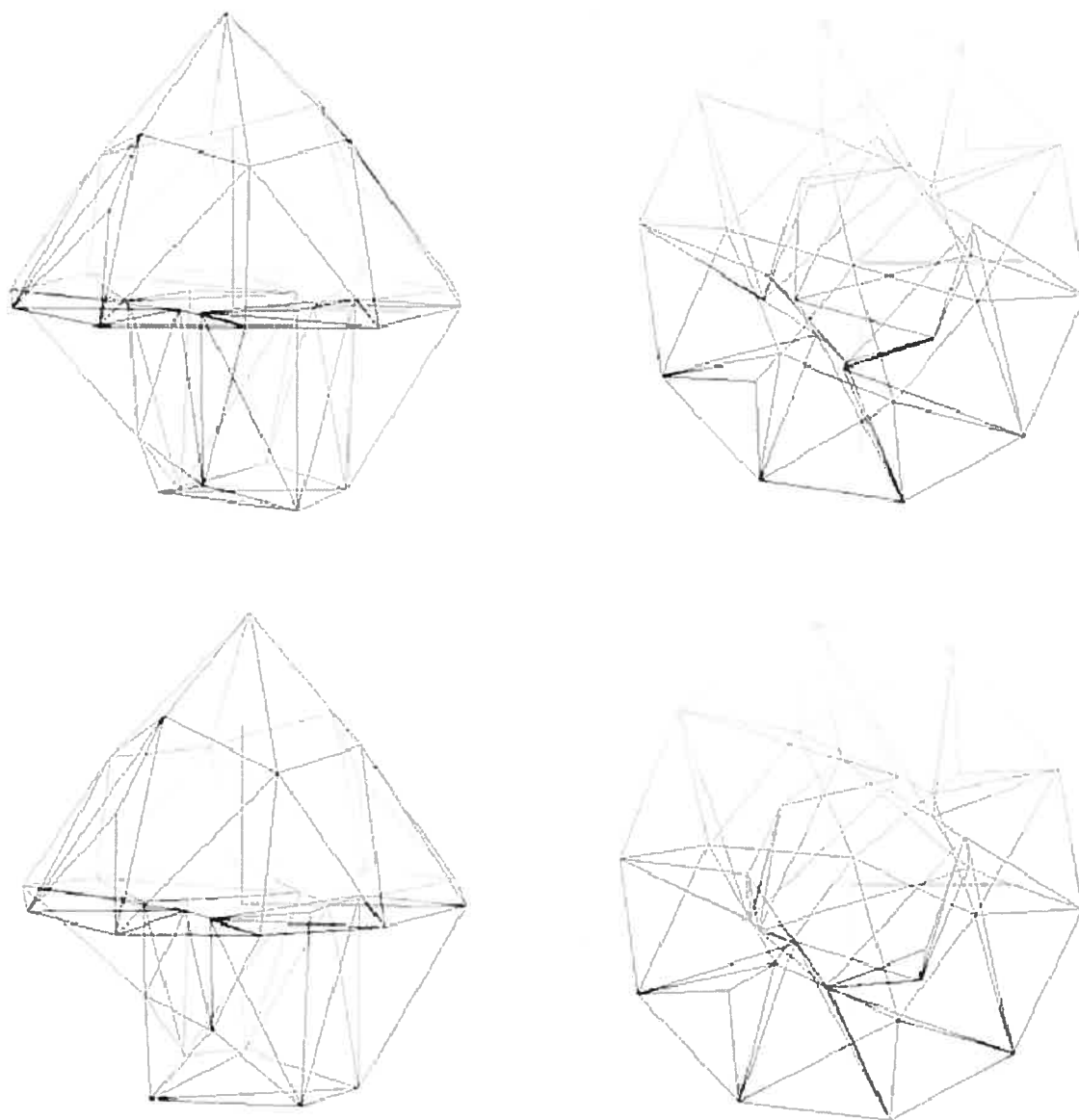


FIG. 4.17 – Images du haut : vue : 0 0 90 0 0 90, énergie : 2.05, images du bas :
vue : 0 0 90 0 0 90 45 0, énergie : 8.1

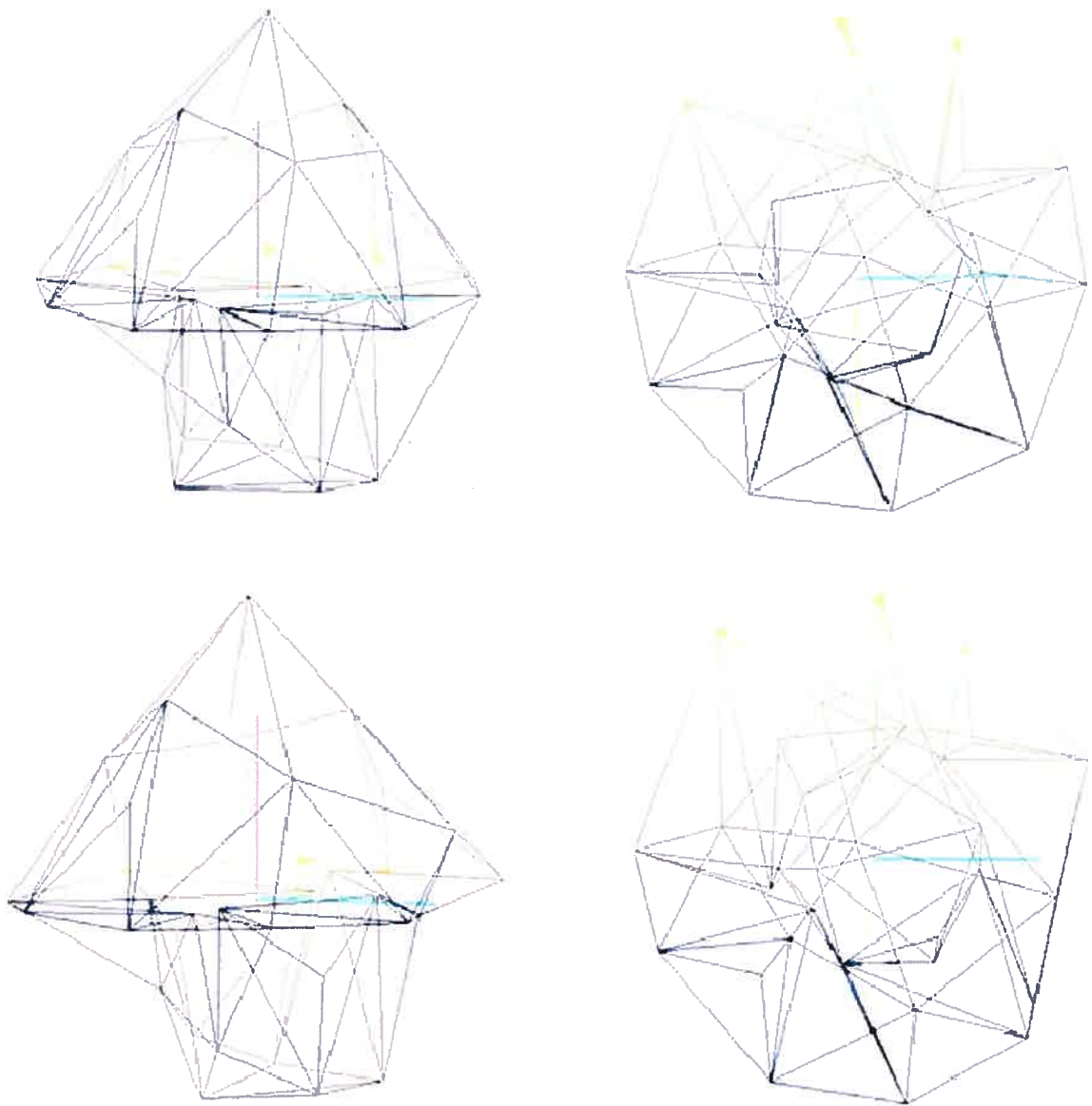


FIG. 4.18 – Images du haut : vue 0 0 90 0 0 90 45 0 0 45, énergie : 9.4, images du bas : vue : 0 0 90 0 0 90 45 0 0 45 20 40 30 20 80 15 30 10, énergie : 30

Dans ces deux derniers tests, le terme de comparaison continue d'être élevé en raison de la complexité de la cohérence entre les différentes vues. Le résultat quant à lui n'a pas vraiment changé et garde la même qualité qu'avec deux vues (voir figure 4.18).

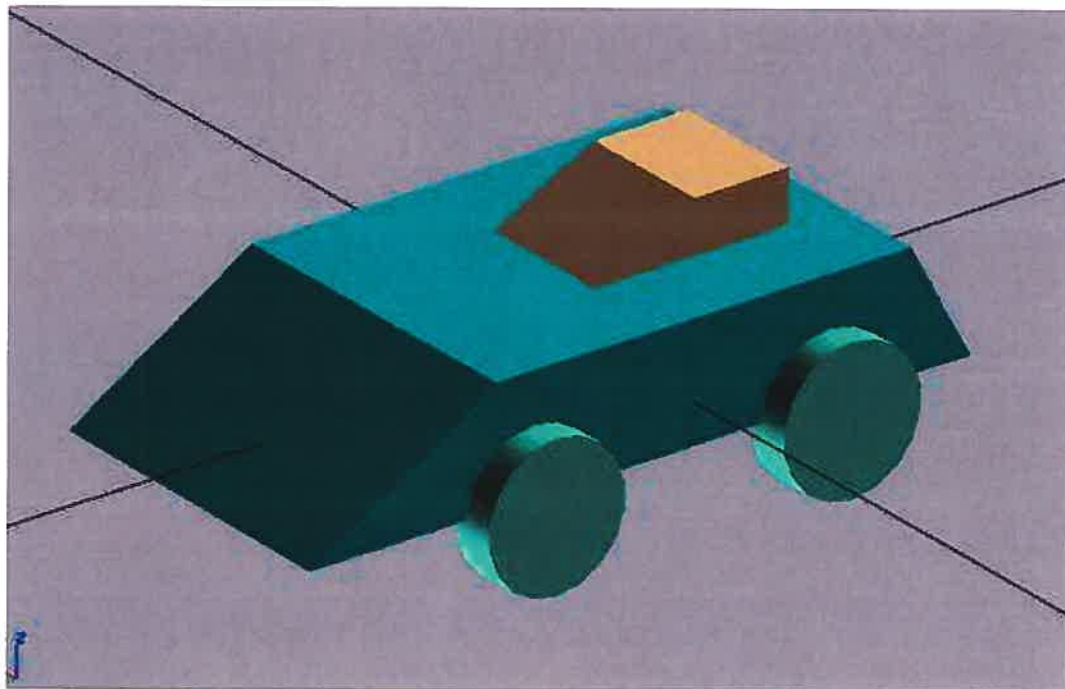


FIG. 4.19 – Voiture : forme cible

4.5 Tests sur des formes complexes

Si on passe à des formes plus complexes les résultats deviennent très approximatifs. Nous faisons ici des tests avec une voiture (voir figure 4.19, 4.20, 4.21, 4.22). On retrouve la forme globale de la voiture. Les principaux problèmes se situent au niveau des parties qui ressortent trop de la forme. On voit en particulier que les roues n'apparaissent presque pas. Un test avec plus de points ne donne pas de résultat plus convainquant.

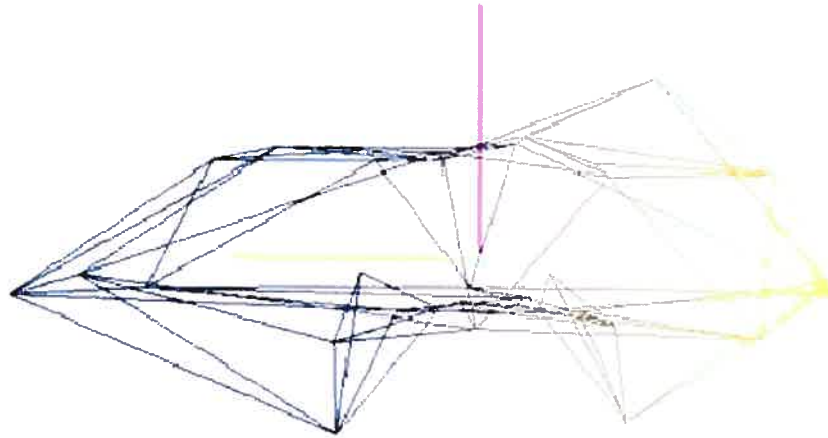


FIG. 4.20 – Voiture : vue de profil

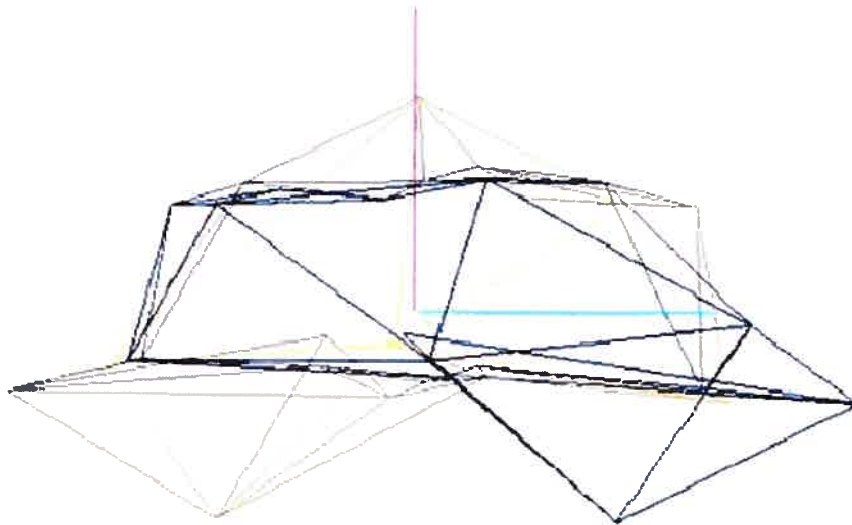


FIG. 4.21 – Voiture : vue de face

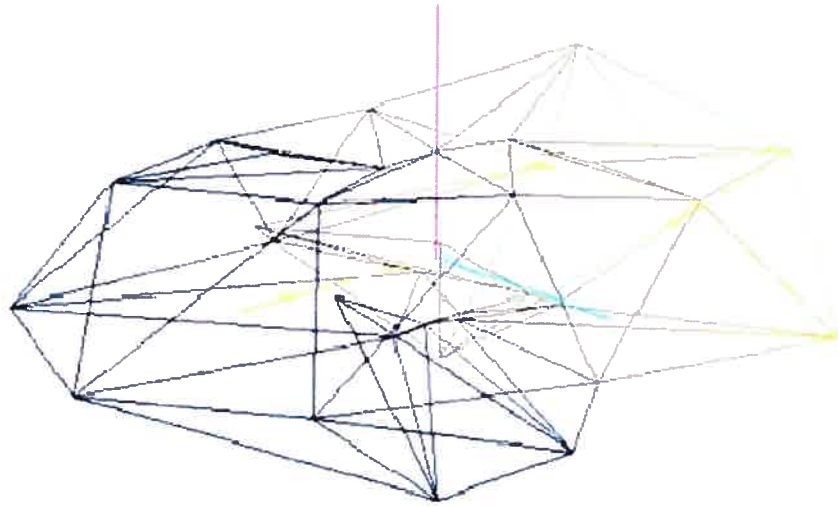


FIG. 4.22 – Voiture : vue oblique Voiture vues 90 0 0 90 méthode 0 itérations 200 pas 0.01

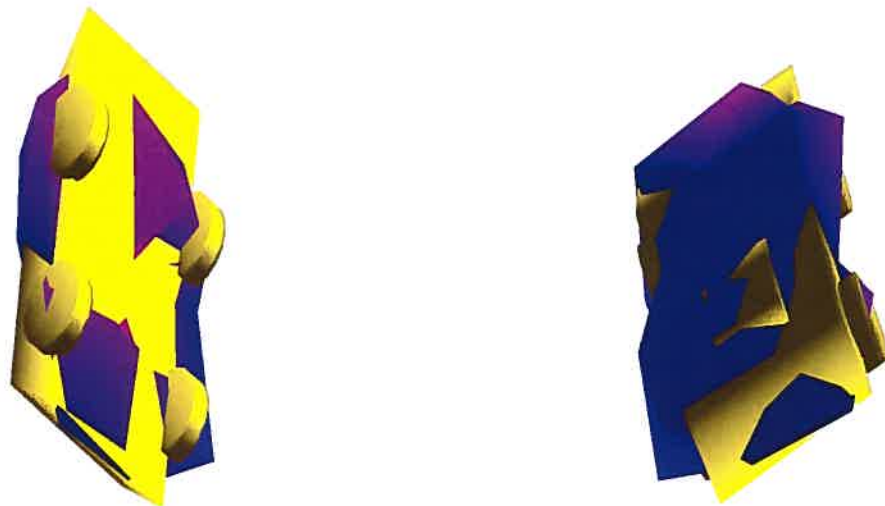
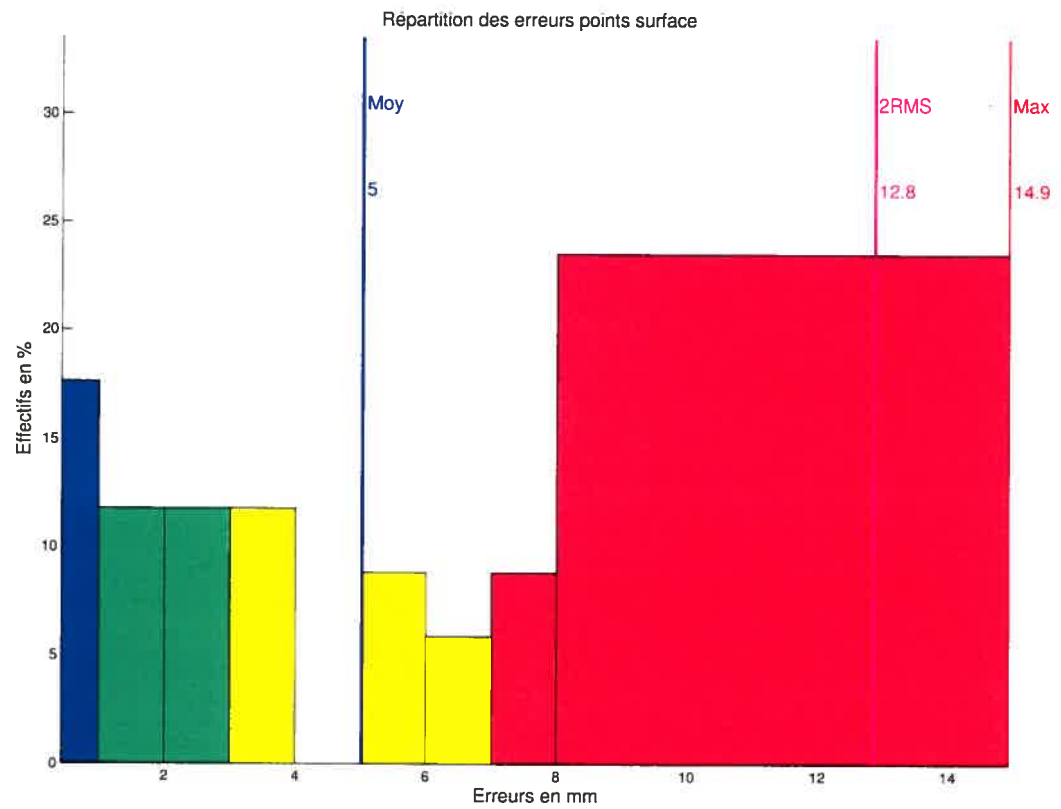


FIG. 4.23 – Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.

Montrons encore un résultat obtenu sur une forme quelconque présentant plusieurs reliefs (voir figure 4.24). On retrouve les principales caractéristiques de la forme, les deux « jambes » en bas, le resserrement au centre, les creux au niveau de la « base des bras » et enfin le haut du « buste » (voir figure 4.24). Les erreurs ne sont pas trop fortes (voir figure 4.25).

Dès lors que l'on veut faire des tests avec des formes comportant des détails, ces derniers apparaissent rarement. Seule la forme globale se dessine. Les parties trop petites qui ressortent n'apparaissent pas.

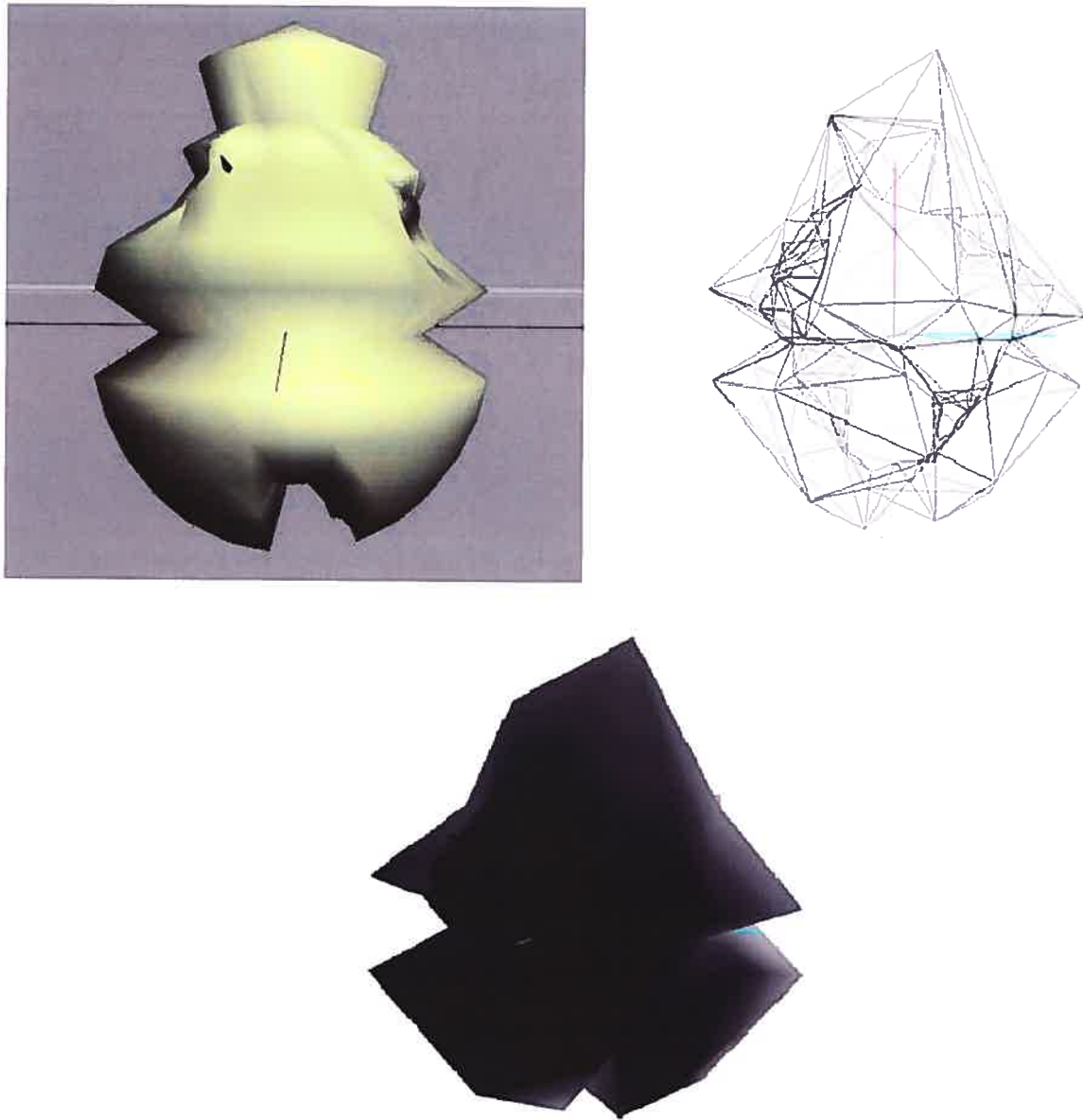


FIG. 4.24 – Corps vues 90 0 0 90 méthode 0 itérations 200 pas 0.01

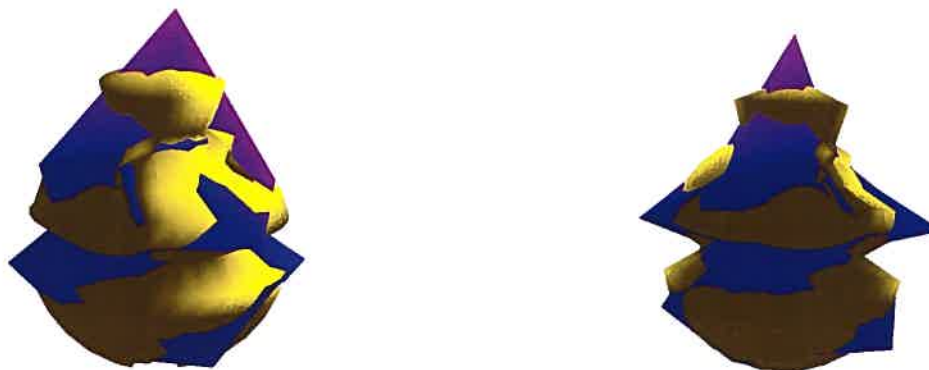
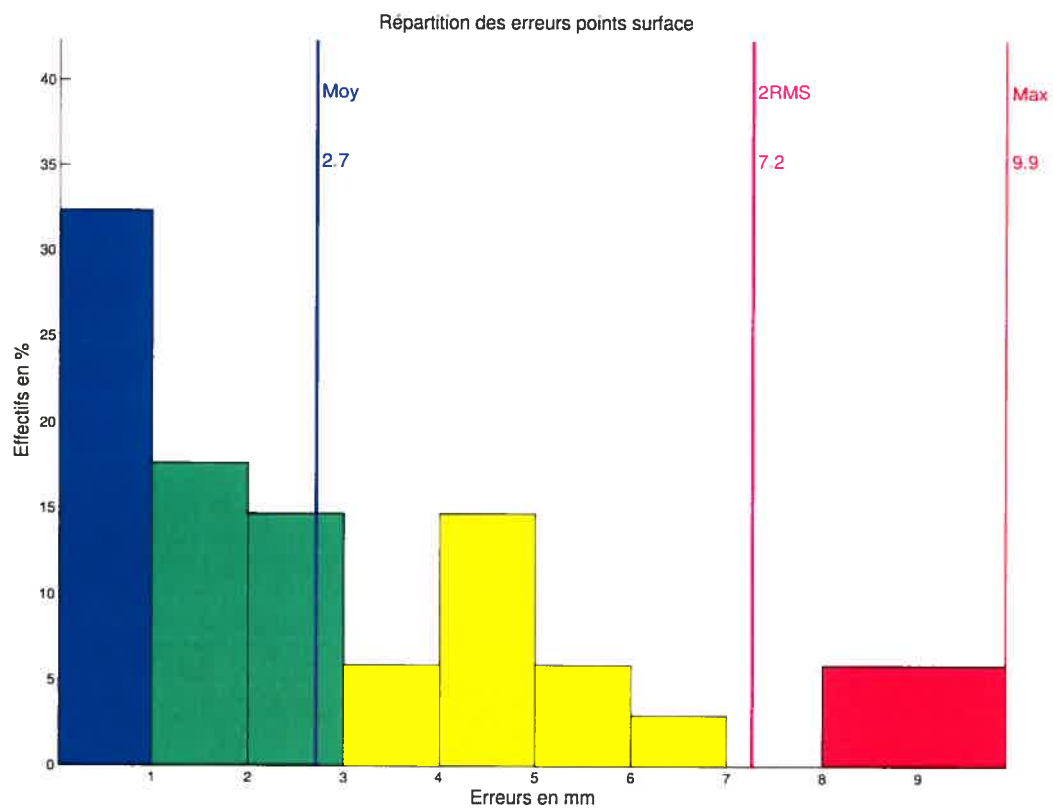


FIG. 4.25 – Diagramme de répartition d'erreurs et superposition de l'image de référence et de l'image obtenue.

1 vue	64 faces	20''	1 vue	256 faces	1'41''
2 vues	64 faces	48''	2 vues	256 faces	2'37''
3 vues	64 faces	1'23''	3 vues	256 faces	3'36''
4 vues	64 faces	2'01''	4 vues	256 faces	4'44''

FIG. 4.26 – Durée de calcul pour effectuer 200 itérations avec un nombre de vues et de faces variable réalisé avec un AMD Athlon XP 2800+.

4.6 Temps de calcul

Terminons enfin ce chapitre par un tableau présentant les temps de calcul nécessaires à l'algorithme. Pour aborder ce sujet, il faut bien distinguer deux points : le temps pris pour effectuer une itération, et le nombre d'itérations effectuées. Le temps pris pour une itération est ce qu'il y a de plus général. Il ne dépend que du nombre de points avec lesquels on travaille et du nombre de vues. Il est donc indépendant du reste et en particulier, il ne dépend pas de la forme ni du pas utilisé pour le déplacement des points. Le nombre d'itérations nécessaires a moins de sens car il dépend de la forme (convergence plus ou moins rapide) de la précision recherchée et du pas utilisé. Plus le pas est petit plus les points vont se déplacer lentement et plus il faudra un grand nombre d'itérations. On peut même généraliser en indiquant que si on divise le pas par deux, il faudra alors multiplier le nombre d'itérations par deux. Pour 64 faces, nos expériences ayant été faites avec 200 itérations et un pas de 0.01, nous donnerons donc les temps réalisés avec ces critères. Nous donnerons également quelques mesures avec 256 faces.

CHAPITRE 5

DISCUSSION ET CONCLUSION

A partir des différents jeux d'essais étudiés dans la partie précédente nous avons donc pu mettre en valeur d'une part le degré de qualité de cette méthode et d'autre part les problèmes qu'elle rencontre dès lors que les formes se complexifient un peu trop.

Intéressons-nous tout d'abord à l'acquisition des données. Celle-ci demande donc différentes cartes de profondeurs obtenues de façon visuelle et en prenant à chaque fois pour chaque direction une vue avant et une vue arrière. La façon la plus simple et la plus classique d'obtenir de telles données est de procéder par stéréovision. En comparaison à de nombreuses méthodes, celle-ci est relativement simple et peu coûteuse. Si on prend par exemple l'acquisition par rayons X , celle-ci demande des instruments de mesure relativement chers. Il faut des scanners de grande précision, et il est beaucoup plus délicat d'obtenir des données non bruitées. De plus l'objet ou la forme étudié doit être suffisamment peu volumineux pour pouvoir être placé à l'intérieur du scanner. La prise de photos n'a de limite que la taille de l'espace où sont faites les mesures. On n'a ainsi aucune limite à l'extérieur. Enfin les radiographies réalisées par radiation sont très longues, de l'ordre de plusieurs heures alors qu'elles sont instantanées lors de la prise de photo. Il reste ensuite le temps de calcul mais celui-ci est également négligeable. Un désavantage cependant flagrant de cette méthode est qu'elle se base sur des données visuelles ce qui lui ferme certains domaines d'application. On peut en particulier citer le domaine médical où l'on a besoin de récupérer des données sur des organes ou élé-

ments situés à l'intérieur du corps humain et donc qu'on ne peut voir par de simples photos. L'étude de défauts à l'intérieur de matériaux tel que des bulles d'air, est également impossible avec notre méthode.

En ce qui concerne la méthode elle-même, l'une de ses grandes qualités est d'être peu coûteuse en information, il suffit en général d'avoir seulement trois vues pour obtenir des résultats optimaux. Comme on a pu le voir précédemment, pour des formes simples, le résultat est assez fidèle à la forme originale. Le résultat est encore appréciable pour des formes assez irrégulières mais dont aucune partie ne ressort trop du corps principal. Précisons aussi que, pour les détails dont seule la taille de ceux-ci poserait problème, il suffirait alors d'augmenter la précision des cartes de profondeurs en relevant plus de points c'est-à-dire en réduisant la taille des mailles de la grille de projection. En revanche, c'est lorsque des parties font apparaître trop d'occlusions qu'apparaissent les limites de notre méthode. Le choix des vues peut par ailleurs avoir une grande importance. Comme nous l'avons vu, les parties vides ou non visibles ne sont pas comptées dans les projections et entraînent donc des incohérences dans les données. Plus les points de vue auront des parties cachées plus le résultat sera faussé.

Ceci nous amène donc à parler d'un axe possible d'évolution de notre méthode. Si l'on réfléchit au choix des vues, on peut faire ressortir plusieurs critères. Le nombre de celles-ci et les directions dans lesquelles elles sont prises. Ces deux facteurs sont liés car une vue peut être incohérente avec certaines et pas avec d'autres. De plus il faut souvent mieux avoir par exemple deux vues avec des directions éloignées qu'un plus grand nombre de vues mais toutes rapprochées. Un point de départ pour un algorithme de recherche de vue optimal pourrait être de travailler sur les

valeurs des cartes de projections. Si l'on considère par exemple que toute forme peut être globalement représentée par un ensemble de surfaces planes et qu'une face sera d'autant plus détectée qu'elle sera parallèle à la direction de la vue, on pourra alors chercher les cartes de projection avec les plus grands écarts de valeurs entre deux zones rapprochées. Ceci n'est qu'une ébauche d'idée et le problème est sans doute très complexe. Une autre idée qui pourrait être intéressante serait de fixer des poids différents aux projections en fonction des vues. Ainsi une vue qui pourrait apporter à la fois des valeurs constructives et d'autres incohérentes pourraient avec un faible poids être tout de même bénéfique.

BIBLIOGRAPHIE

- [BI99] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33(3) :181–200, 1999.
- [BMDG05] S. Benameur, M. Mignotte, F. Destrempes, and J.A. De Guise. 3d biplanar reconstruction of scoliotic rib cage using the estimation of a mixture of probabilistic prior models. *IEEE Transactions on Biomedical Engineering*, 52(10) :1713–1728, 2005.
- [BMLG05] S. Benameur, M. Mignotte, H. Labelle, and J.A. De Guise. A hierarchical statistical modeling approach for the unsupervised 3d biplanar reconstruction of the scoliotic spine. *IEEE Transactions on Biomedical Engineering*, 52(12) :2041–2057, 2005.
- [CHRM96] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs. A maximum likelihood stereo algorithm. *CVIU*, pages 542–567, 1996.
- [GLY92] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. *ECCV*, pages 425–433, 1992.
- [IG93] H. Ishikawa and D. Geiger. A parallel stereo algorithm that produces dense depth maps and preserves image features. 6 :35–49, 1993.
- [IG98] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. *ECCV*, pages 232–248, 1998.
- [KS00] K. Kutulakos and S. Seitz. A theory of shape by space carving. *IJCV*, 38(3) :199–218, 2000.
- [Kut00] K. Kutulakos. Approximate n-view stereo. *ECCV*, 1 :67–83, 2000.
- [OK85] Y. Ohta and T. Kanade. Stereo by intra- and interscanline search using dynamic programming. *IEEE TPAMI*, 7(2) :139–154, 1985.

- [RC98] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *ICCV*, pages 492–499, 1998.
- [SCD⁺06] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR*, 1 :219–528, 2006.
- [SCMS99] G. Slabaugh, B. Culbertson, T. Malzbender, and M. Stevens. Shape reconstruction in projective grid space from large number of images. *CVPR*, 2 :49–54, 1999.
- [SCMS04] G. Slabaugh, B. Culbertson, T. Malzbender, and M. Stevens. Methods for volumetric reconstruction of visual scenes. *IJCV*, 57(3) :179–199, 2004.
- [SD97] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. *CVPR*, pages 1067–1073, 1997.
- [SMD04] C. Soussen and A. Mohammad-Djafari. Polygonal and polyhedral contour reconstruction in computed tomography. *IEEE transaction on image processing*, 13, 2004.
- [SSH02] G.G. Slabaugh, R.W. Schafer, and M.C. Hans. Multi-resolution space carving using level set methods. *ICIP*, 2 :545–548, 2002.
- [SSZ01] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *SMBV*, pages 131–140, 2001.
- [THS04] A. Treuille, A. Hertzmann, and S. Seitz. Example-based stereo with general brdfs. *ECCV*, 2 :457–469, 2004.
- [Vek99] O. Veksler. *Efficient Graph-based Energy Minimization Methods in Computer Vision*. PhD thesis, 1999.