

Institut National Polytechnique de Grenoble

---

# Reconstruction 3D de l'enveloppe photométrique à l'aide d'une discrétisation adaptée.

---

Julien Morat

DEA Imagerie, Vision, Robotique

Année 2002 - 2003

Laboratoire GRAVIR (CNRS, INPG, INRIA, UJF)

Equipe Movi

Responsables de recherche : Edmond Boyer



Je tiens à remercier Edmond Boyer pour le temps et la confiance qu'il m'a accordé, ainsi que pour ses excellents conseils.

Je remercie également Messieurs Jim Crowley et Peter Sturm de présider mon jury ainsi que Monsieur Jean-Dominique Gascuel pour avoir accepté d'en être membre.

Enfin, je remercie Bebert, Matthieu, Remy, Caroline, Cédric, Fabien, Pascal, Laurent, Pierre, Olivier, et tous les autres pour leur aide, leur bonne humeur, bref, tout ce qui a fait l'excellente ambiance de ce DEA.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>6</b>
<b>2</b>	<b>Définitions</b>	<b>8</b>
2.1	Géométrie projective . . . . .	8
2.2	Le modèle de caméra . . . . .	9
2.2.1	Le modèle sténopé . . . . .	9
2.2.2	Matrice de projection (calibration) . . . . .	9
2.3	Géométrie épipolaire . . . . .	9
2.3.1	Droite épipolaire . . . . .	9
2.3.2	Epipole . . . . .	10
2.3.3	Matrice fondamentale . . . . .	10
2.4	Cohérence photométrique . . . . .	11
2.4.1	Critères de cohérence photométrique . . . . .	11
2.4.2	Espace de couleur . . . . .	11
2.5	Enveloppe visuelle . . . . .	12
2.6	Enveloppe photométrique . . . . .	12
<b>3</b>	<b>Etat de l'art</b>	<b>14</b>
3.1	Synthèse d'un modèle volumique . . . . .	14
3.1.1	Espace discret . . . . .	14
3.1.2	Algorithmes de sculpture (Voxel Coloring) . . . . .	15
3.2	Synthèse d'un modèle surfacique . . . . .	17
3.2.1	Surface polyédrique . . . . .	17
3.2.2	Surface du second ordre . . . . .	18
3.3	Synthèse d'une nouvelle vue . . . . .	18
3.3.1	Enveloppe visuelle par une approche image ( <b>IBVH</b> ) . . . . .	18
3.3.2	Enveloppe photométrique par une approche image . . . . .	19
<b>4</b>	<b>Extraction d'informations clés dans l'image</b>	<b>21</b>
4.1	Algorithmes de segmentation image . . . . .	21
4.1.1	Blobworld [CTB <sup>+</sup> 99] . . . . .	21
4.1.2	Edgeflow . . . . .	22
4.2	Discretisation des contours . . . . .	23
4.2.1	Polygonalisation naïve . . . . .	23
4.2.2	Polygonalisation optimisée . . . . .	23

<b>5</b>	<b>Calcul de points sur la surface de l'enveloppe photométrique</b>	<b>24</b>
5.1	Création des points sur le contour . . . . .	24
5.2	Création des segments épipolaires . . . . .	25
<b>6</b>	<b>Discretisation volumique</b>	<b>27</b>
6.1	Triangulation de Delaunay . . . . .	27
6.2	Les limites de la triangulation . . . . .	27
<b>7</b>	<b>Sculpture</b>	<b>29</b>
7.1	Approche . . . . .	29
7.2	Cohérence photométrique . . . . .	30
7.2.1	Approximation du tétraèdre . . . . .	30
7.2.2	La mesure de distance . . . . .	30
7.3	Test de visibilité . . . . .	31
7.3.1	Visibilité d'un tétraèdre . . . . .	31
7.3.2	Visibilité d'une face d'un tétraèdre . . . . .	32
7.4	Ordre de parcours des tétraèdres . . . . .	32
7.4.1	Parcours via le <i>plane sweep</i> . . . . .	33
7.4.2	Liste des tétraèdres de surface . . . . .	33
7.5	Sculpture dans la triangulation . . . . .	34
<b>8</b>	<b>Résultats</b>	<b>36</b>
8.1	Apport de la photométrie . . . . .	36
8.2	Apport de la discrétisation adaptée . . . . .	37
8.3	Complexité . . . . .	37
8.3.1	Triangulation imparfaite . . . . .	38
<b>9</b>	<b>Conclusion et perspectives</b>	<b>40</b>
9.1	Améliorer les différentes étapes . . . . .	40
9.2	Une meilleur approximation des faces . . . . .	40
9.3	Diminuer le nombre de points 3D initiaux . . . . .	41
9.4	Application à des images réelles . . . . .	41

# Chapitre 1

## Introduction

La numérisation d'objets réels débouche sur de nombreuses applications. Comme intégrer des acteurs dans des studios virtuels, modéliser des décors et des personnages pour le cinéma ou les jeux vidéo, ...

A l'heure actuelle, les techniques les plus avancées sont des méthodes invasives. La robotique propose des scanners basés sur des palpeurs. Un bras palpe la surface d'un objet réel pour en reconstituer la surface. La vision par ordinateur propose elle aussi des techniques de scanner, tel que les scanners lasers. Mais ces techniques atteignent très vite leurs limites. Elles requièrent la mise en place d'un matériel cher, difficile à utiliser. Les perpétuelles avancées dans le domaine de la vision par ordinateur apportent de nouvelles solutions de plus en plus fiables, souples et simples à utiliser.

Le problème de reconstruction reste un problème de vision par ordinateur difficile. D'autant qu'il n'existe souvent pas de solution unique. Le problème de reconstruction peut être envisagé comme un problème d'optimisation sous les contraintes des images. [Lau94] définit une approximation de la solution sous le terme *d'enveloppe visuelle*. L'enveloppe visuelle est la solution qui maximise le volume sous les contraintes des seuls contours. L'objet est strictement inclus à l'intérieur de cette enveloppe. Elle ne comporte pas de concavités. [SD97] [CMS99] [Dye01] [SS99], y ajoutent les contraintes photométriques (couleur) afin d'obtenir l'enveloppe photométrique. Les contraintes photométriques étant plus fortes, la reconstruction est plus précise. Elle permet notamment de rendre les concavités de la surface.

La taille des données et la complexité du problème ne permettent pas encore d'envisager de le résoudre sous la forme d'une optimisation sous contraintes. Les techniques actuelles ont une approche algorithmique. Trois types d'approches basées sur des modèles s'affrontent : volumique, surfacique, et image.

Les approches volumiques s'appuient sur une discrétisation de l'espace en éléments de base, les voxels (sorte de pixel 3D). Chaque voxel possède la propriété de contenir de la matière, c'est à dire d'être cohérent. Le processus de reconstruction consiste à évaluer la cohérence de chaque voxel. La discrétisation de l'espace est le plus souvent régulière. Mais il existe des alternatives, comme par exemple les **octrees** ou encore **projective gride space**. Les premières approches reconstruisent l'enveloppe visuelle. Par la suite, plusieurs méthodes de "voxel carving" reprirent le principe pour l'étendre aux enveloppes photométriques.

Les approches surfaciques abordent le problème sous un autre aspect. Les limites de l'objet, points et facettes, sont calculées directement depuis les informations images. L'enveloppe visuelle est calculée par l'intersection des cônes visuels, directement issus des contours. Les premières approches s'appuyaient sur une approximation polygonal du contour. D'autres approches plus récentes comme [LBP01] reconstruisent des points isolés par l'approximation au second ordre de la surface. Les reconstructions surfaciques sont plus précises que les reconstructions volumiques. Ceci est essentiellement dû au fait que les reconstructions surfaciques ne souffrent pas d'une discrétisation arbitraire. Cependant, la surface du modèle produit est souvent incomplète. Une des raisons à cela est que les intersections des cônes visuels est souffrent souvent d'erreurs d'approximation.

Les approches images sont très différentes des deux précédentes. Le but premier d'une approche image n'est pas de reconstruire la géométrie de l'objet, mais seulement de synthétiser une nouvelle vue de celui-ci. Les techniques et applications ne sont pas les mêmes non plus. La reconstruction image fournira des meilleurs résultats visuels. Mais, inconvénient majeur, l'information géométrique reconstituée n'est que partielle, et dépend du point de vue. Pour chaque nouveau point de vue, la reconstruction doit être calculée à nouveau. Les techniques de reconstruction image sont basées sur l'analyse des lignes de vue du point de vue virtuel. Pour qu'un point se projette dans une vue, il doit être le plus proche le long du rayon de projection.

Dans le cadre de ce projet, nous explorons de nouvelles méthodes hybride pour reconstruire l'enveloppe photométrique d'une scène complexe. Notre approche est basée sur une généralisation des approches déjà existantes. De manière similaire à [BF03], elle tire avantage à la fois de la robustesse des approches volumiques et de la précision des approches surfaciques. Le principe fondamental reste un algorithme du type **voxel coloring**. La discrétisation de l'espace est toujours nécessaire. Mais, au contraire de la plus part des méthodes, les cellules élémentaires ne sont pas définies arbitrairement, mais à partir d'informations images, comme les contours.

Nous allons présenter la méthode et ces apports en quatre axes. Tout d'abord, nous verrons comment extraire les informations de contour et de région des images. Ensuite, nous verrons comment exploiter ces informations, pour en extraire des points situés sur la surface de l'enveloppe photométrique. Une triangulation de Delaunay de ces points permet de discrétiser l'espace. Enfin, nous exposerons comment sculpter l'enveloppe photométrique dans l'ensemble des tétraèdres formés par la triangulation. Ces derniers étant obtenus par une triangulation de Delaunay des points caractéristiques calculés. En dernière partie, nous présenterons les résultats expérimentaux obtenus. Nous comparerons les apports et les limites de la méthode par rapport aux approches conventionnelles.

# Chapitre 2

## Définitions

La reconstruction est un domaine particulier de la vision par ordinateur. Certaines notions sont des outils indispensables, comme la géométrie projective, le modèle sténopé ou la géométrie épipolaire. C'est pourquoi nous faisons ici quelques rappels succincts de ces notions.

### 2.1 Géométrie projective

Rappelons que le système de coordonnées homogènes utilisé en géométrie projective comporte  $n + 1$  coordonnées, définies à un facteur d'échelle près. Elles représentent, en général, un vecteur de l'espace euclidien de dimension  $n$ . Par habitude, un point d'un espace 3D sera noté  $X$  (majuscule). Ses coordonnées homogènes comportent quatre composantes. En 2D, les points et lignes seront notés  $x$  et  $l$  (minuscules). En coordonnées homogènes, ils seront représentés par 3 composantes.

L'égalité projective à un facteur d'échelle près entre deux expressions  $A$  et  $B$  sera notée

$$A \sim B$$

Le vecteur 3D euclidien correspondant au coordonnées homogènes  $X$  sera noté  $\bar{X}$  de sorte que :

$$X \sim \begin{pmatrix} \bar{X} \\ 1 \end{pmatrix}$$

Le produit scalaire entre 2 vecteurs  $X$  et  $Y$  sera noté  $X^T \cdot Y$ . De la même manière que ci-dessus on notera  $\bar{x}$  le vecteur euclidien de dimension 2 tel que

$$x \sim \begin{pmatrix} \bar{x} \\ 1 \end{pmatrix}$$

Nous rappelons à cette occasion que l'intersection de 2 droites  $l_1$  et  $l_2$  est le point  $x$  tel que  $x \sim l_1 \wedge l_2$  avec  $\wedge$  le produit vectoriel. De même la droite passant par 2 points  $x_1$  et  $x_2$  est la droite  $l$  avec  $l \sim x_1 \wedge x_2$ ; en outre une droite projective  $l$  passe par le point  $x$  ssi  $x^T \cdot l = 0$ .



## 2.2 Le modèle de caméra

### 2.2.1 Le modèle sténopé

Pour résoudre numériquement et géométriquement les problèmes liés à la reconstruction, nous avons besoin d'un modèle qui décrit comment les objets 3D se projettent sur une image 2D. Le modèle le plus répandu est le modèle dit **sténopé** (ou modèle de trou d'épingle). Il n'est pas complet. Il ne tient par exemple pas en compte des distorsions. Cependant, il est suffisamment proche du modèle physique et séduisant de simplicité. Le modèle sténopé est décrit par l'ensemble d'un centre de projection  $c$  et d'un plan image. Un point 3D  $P$  du monde se projette sur le plan image en  $p$  à l'intersection du plan image et de la droite  $(cP)$  formée du centre de projection et du point 3D  $P$ . Cette droite  $(cP)$  est appelée ligne de vue ou rayon de projection.

### 2.2.2 Matrice de projection (calibration)

La **matrice de projection**  $P_i$  est la matrice  $4 \times 3$  tel que  $q_i \sim P_i \cdot Q$  où  $Q$  est un point du monde 3d, et  $q_i$  sa projection dans l'image  $i$ .

$$P \sim \begin{pmatrix} K & 0 \end{pmatrix} \begin{pmatrix} R & -Rt \\ 0^T & 1 \end{pmatrix} \quad (2.1)$$

$$\begin{pmatrix} KR & -KRt \end{pmatrix} \quad (2.2)$$

$$\begin{pmatrix} \bar{M} & m \end{pmatrix} \quad (2.3)$$

Où  $R$  est la matrice de rotation de taille  $3 \times 3$  et  $t$  le vecteur de translation de taille 3.

## 2.3 Géométrie épipolaire

### 2.3.1 Droite épipolaire

La géométrie épipolaire est la relation qui lie deux images prises de point de vues différents. Cette transformation n'est pas bijective. C'est à dire qu'à un point dans une image, n'est pas associé un unique point dans l'autre. Pourtant, étant donné la position d'un point dans une image, la position du point correspondant dans l'autre image n'est pas quelconque. Elle est contrainte par une droite (cf. figure 2.1 page 10). Cette droite est la droite épipolaire.

Nous savons que le point 3D  $Q$  se projette en  $q_1$  dans la première image, et en  $q_2$  dans la deuxième image.  $q_1$  est connu. Premièrement, que peut-on dire de la position de  $Q$ ? Il se projette en  $q_1$ , il appartient donc au rayon de projection passant par  $q_1$  (et par le centre de projection). Les caméras ayant été calibrées, ce rayon est défini. Il se projette en une droite  $l_2$  dans la deuxième image. Ainsi, la projection du point 3D  $Q$  se trouve sur cette droite. C'est ainsi que se formalise la contrainte épipolaire. Pour plus de détails, voir le cours DEA IVR [Stu03].

### 2.3.2 Epipole

Toutes les lignes de vue d'une même caméra se coupent en un point. En effet, par définition (c.f 2.2.1) toutes les lignes de vues passent par le centre optique de la caméra. De ce fait, les droites épipolaires correspondantes se couperont toutes en un même point. Il est aisé de constater que ce point n'est autre que la projection du centre de projection de la première caméra dans la deuxième (cf 2.1 ). Ce point est l'épipole.

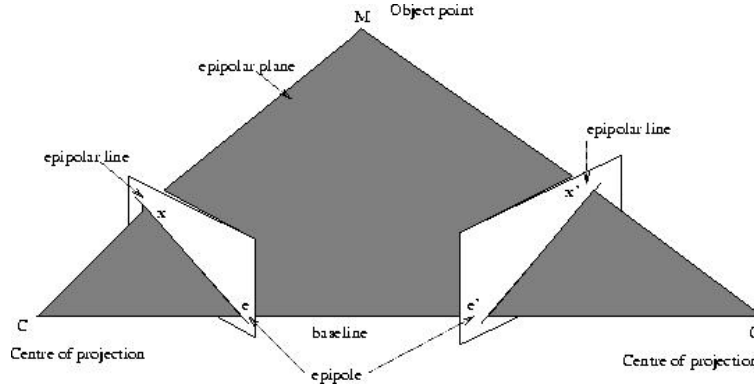


FIG. 2.1 – Modèle sténopé.

### 2.3.3 Matrice fondamentale

La matrice fondamentale est un moyen de représenter la contrainte épipolaire sous forme algébrique. Soient  $Q$  un point 3D,  $q_1$  et  $q_2$  ses projections dans les images 1 et 2, et  $F_{12}$  la matrice fondamentale liant l'image 1 à l'image 2, alors on peut écrire la contrainte sous la forme :

$$q_2^\top \cdot F \cdot q_1 = 0 \quad (2.4)$$

Avec  $F_{ij}$  la matrice fondamentale liant l'image  $i$  à l'image  $j$ .

$$F_{ij} \sim (K_j R_j)^T [t_i - t_j] \wedge (R_i^T K_i^{-1}) \quad (2.5)$$

$$F_{ij} \sim (K_j R_j)^T [t_i - t_j] \wedge ((K_i R_i)^{-1}) \quad (2.6)$$

$$F_{ij} \sim \overline{M}_i^T [t_i - t_j] \wedge \overline{M}_i^{-1} \quad (2.7)$$

$t_i$  et  $t_j$  peuvent facilement se calculer :

$$-KRt = m \quad (2.8)$$

$$KRt = -m \quad (2.9)$$

$$t = -(KR)^{-1}m \quad (2.10)$$

$$t = -\overline{M}^{-1}m \quad (2.11)$$

La matrice fondamentale  $F_{ji}$  permettant de passer de l'image  $j$  à l'image  $i$  est la transposée de  $F_{ij}$ .

## 2.4 Cohérence photométrique

### 2.4.1 Critères de cohérence photométrique

Beaucoup de méthodes de reconstruction sont basées sur le principe de cohérence photométrique. Une scène reconstruite est dite photo-cohérente, si, reprojétée suivant les points de vues de références, les images obtenues sont identiques. (2.4(c) p.13 est cohérent 2.4(a) p.13 ). Les critères de cohérences peuvent être aussi complexes qu'une surface peut l'être. L'illumination d'un point dépend d'un grand nombre de paramètres : sa couleur, son pouvoir réfléchissant, sa transparence, son indice de réfraction, sa normale à la surface, ... [SD97] démontrèrent que le critère de cohérence couleur (ou photométrique) suffit pour reconstruire convenablement la majorité des objets. Les scènes naturelles étant souvent lambertiennes, le critère photométrique suffit.

### 2.4.2 Espace de couleur

Pour évaluer la cohérence couleur, une mesure de distance entre deux couleurs est nécessaire. Pour évaluer une distance, nous considérons une couleur comme un point dans un espace à trois dimensions. La représentation la plus courante est la décomposition suivant les trois canaux de couleurs primaires : rouge, vert et bleu (cf. 2.2). Une couleur est alors représentée comme un point dans le cube  $RGB$  (voir figure 2.2).

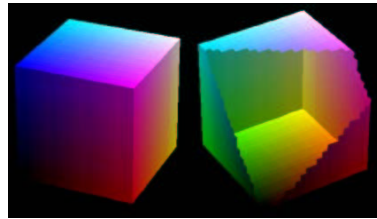


FIG. 2.2 – Le cube représentant l'espace RGB, dont les trois composantes sont le rouge, le vert et le bleu.

Dans le cas de prises de vue réelles, l'image est souvent bruitée. Le bruit peut affecter la qualité des résultats. L'information la plus affectée par le bruit est l'intensité lumineuse. Un simple changement d'orientation peut très fortement modifier l'éclairage de l'image. Pour plus de robustesse, nous avons choisi d'éliminer l'information d'intensité. Nous allons donc considérer uniquement la chrominance du pixel. Elle peut être représentée sur 2 dimensions. Plusieurs espaces séparent explicitement la composante de luminance, de celles de chrominances (HSV, HSL, RGn, YUV, ...). Nous avons choisi l'espace  $RG_{normalis}$ . Il est très facile de convertir une couleur au format  $RGB$  au format  $RG_n$  :

$$R_n = \frac{R}{R + G + B} \quad (2.12)$$

$$G_n = \frac{G}{R + G + B} \quad (2.13)$$

L'espace des couleurs dont nous parlerons par la suite, est l'espace  $RG_n$ . C'est un espace à 2 dimensions.

## 2.5 Enveloppe visuelle

La première reconstruction volumétrique à base d'images date du milieu des années 1980. C'est dans [Lau94] qu'apparut pour la première fois le terme d'enveloppe visuelle (Visuall hull en anglais). L'enveloppe visuelle n'est en fait qu'une approximation de l'objet. C'est le volume convexe englobant l'objet. Elle est déterminée uniquement à l'aide des silhouettes de celui-ci. Au plus le nombre de vues est important, au plus l'enveloppe visuelle sera définie avec précision.

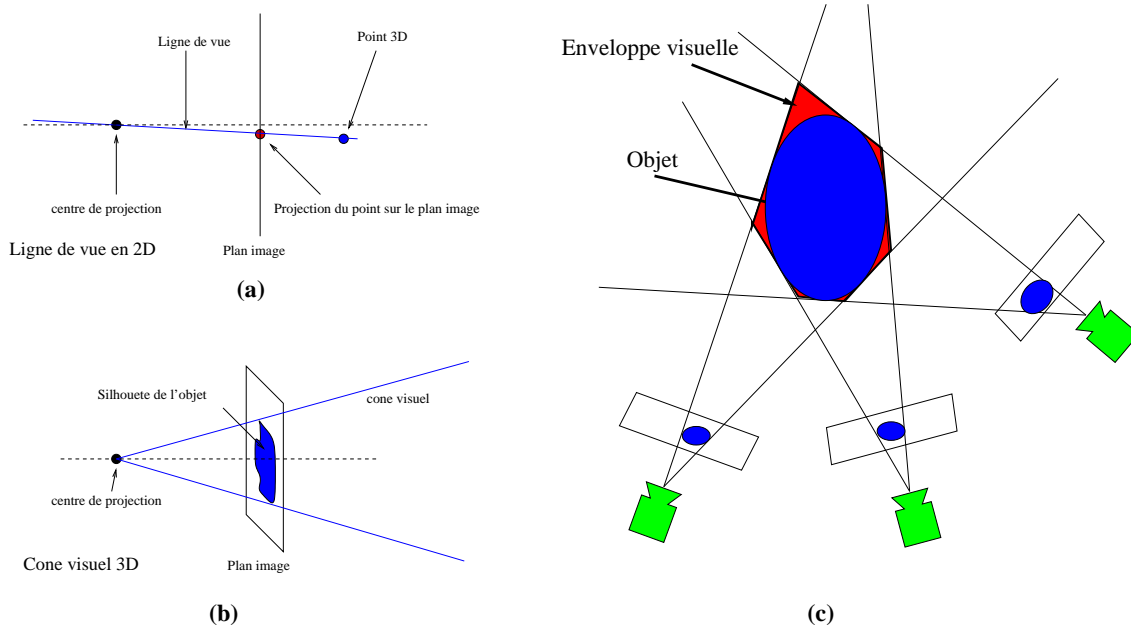


FIG. 2.3 – (a) La ligne de vue associée à un pixel. (b) Le cône visuel associé à une caméra. (c) L'enveloppe visuelle associée à trois caméras.

Dans l'image, chaque pixel est associé à un rayon 3-D qui passe par ce pixel (et par le centre optique 2.3 (a)). Les rayons qui correspondent à des pixels de la silhouette de l'objet « sculptent » un cône généralisé. Ce cône ayant pour section la forme de la silhouette (cf. 2.3 (b) ). L'objet 3-D est entièrement contenu à l'intérieur de cette enveloppe. Avec plusieurs images, nous obtenons plusieurs cônes dont l'intersection donne une surface englobante de la forme de l'objet (cf. 2.3 (c) ).

## 2.6 Enveloppe photométrique

Le concept d'enveloppe photométrique est similaire à celui d'enveloppe visuelle. Nous avons vu que pour reconstruire l'enveloppe visuelle, sert l'unique information de silhouette. En ajoutant l'information photométrique, nous ajoutons des contraintes qui raffinent l'enveloppe. L'enveloppe photométrique est strictement incluse dans l'enveloppe visuelle. Le cas limite étant rencontré pour un objet de couleur uniforme, auquel cas l'enveloppe photométrique correspond exactement à l'enveloppe visuelle. C'est donc la non-uniformité de la couleur qui permet de rendre compte au mieux de la surface, par exemple, de modéliser les

concavités de l'objet.

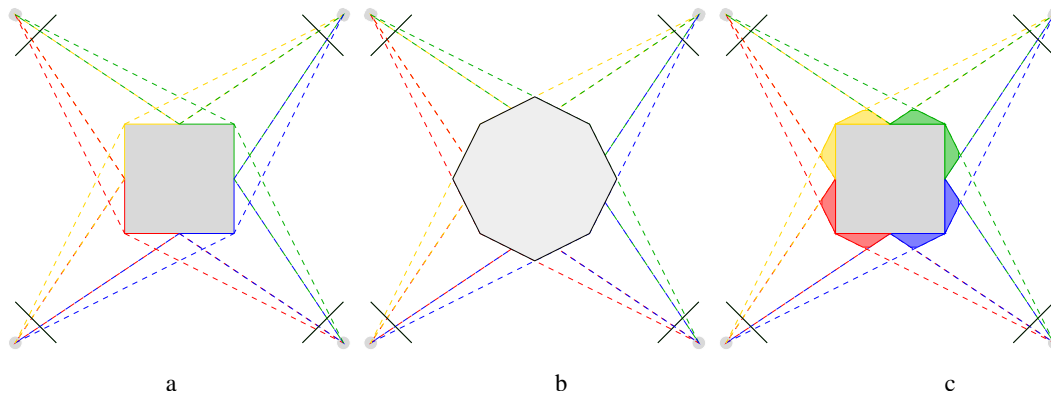


FIG. 2.4 – (a) Le modèle réel. (b) L'enveloppe visuelle. (c) L'enveloppe photométrique.

Comme l'illustre la figure 2.5, un jeu d'image de référence donne lieu à une ambiguïté de solutions. Il en est une qui maximise le volume. [KS98] la définit comme l'enveloppe photométrique maximale. Par abus de langage, nous parlerons de l'enveloppe photométrique pour désigner l'enveloppe photométrique maximale.

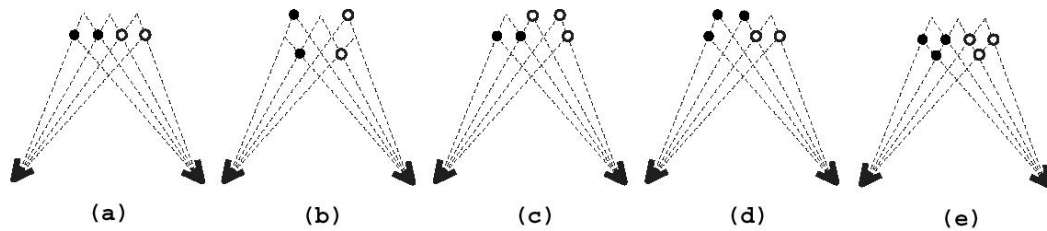


FIG. 2.5 – Ambiguïté de reconstruction. Ces 5 scènes sont toutes cohérentes depuis les 2 points de vue. tiré de [SD97].

## Chapitre 3

# Etat de l'art

Dans ce chapitre, nous allons présenter quelques unes des différentes méthodes de modélisation à base d'images. Les différentes approches abordées diffèrent essentiellement par le modèle utilisé. Le principe de base reste toujours d'utiliser l'information stéréoscopique conjointement à des critères de cohérences, pour recouvrer la troisième dimension. Nous allons présenter les trois principales approches, chacune basée sur un modèle différent.

### 3.1 Synthèse d'un modèle volumique

La représentation par des modèle volumique n'a cessé de croître depuis leur introduction dans les années 70 dans le domaine de l'imagerie médicale. Tous les algorithmes (présentés dans [SCMS]) suivent le schéma du **voxel coloring**. La reconstruction s'effectue par la sculpture de l'objet dans un espace discret.

#### 3.1.1 Espace discret

##### Espace voxélique régulier

Les voxels correspondent à l'unité de l'espace 3D numérique, de la même manière que le pixel représente une unité de l'espace 2D numérique (image). L'ensemble de voxels forme un sous-espace de taille finie, de forme parallélépipédique. S'ils sont répartis régulièrement, les voxels ont la forme de cubes. Dans le processus de reconstruction, ils peuvent prendre deux états : vide ou plein. Ils sont vides s'ils ne contiennent pas de matière, c'est à dire qu'ils ne se trouvent pas à l'intérieur d'un objet. Ils sont pleins sinon.

##### Espace voxélique projectif

Pour éviter les inconvénient liés à la calibration de caméra, [SK] propose un espace discret alternatif : un espace projectif. Au lieu d'un espace composé de voxels cubiques, les voxels ont une forme irrégulière dépendante de la géométrie épipolaire de 2 caméras de références (voir figure 3.2 page 15). La taille des voxels est donc proportionnelle à leur distance des deux caméras. Une fois la grille projective définie, il est facile de projeter un point dans toutes les autres vues.



FIG. 3.1 – Chien modélisé par des voxels à divers niveau de résolution.

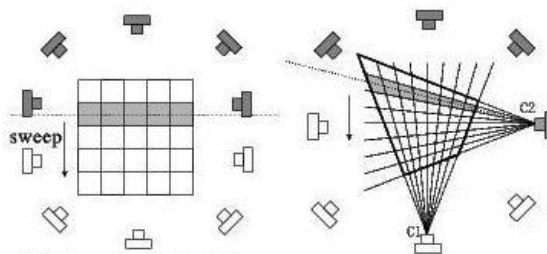


FIG. 3.2 – A gauche : un espace voxélique traditionnel. A droite : un espace voxélique projectif basé sur 2 caméras. Tiré de [SK].

### Volumétrie Warping

Les grilles voxéliques traditionnelles définissent un volume de reconstruction de taille finie. [SMC01] imagine une méthode permettant de s'abstraire cette limitation. Il présente un espace voxélique déformé (voir figure 3.3 page 16), afin que le domaine de reconstruction soit étendu à l'infini. Il est ainsi possible d'intégrer à la reconstruction des objets très éloignés de la caméra, comme le fond ou l'environnement.

### 3.1.2 Algorithmes de sculpture (Voxel Coloring)

Comme nous l'avons vu dans la partie 2.6 (page 12), l'enveloppe photométrique est strictement incluse dans l'enveloppe visuelle. Ce lemme garantit que l'enveloppe photométrique s'obtient uniquement en éliminant des parties de l'enveloppe visuelle. L'algorithme s'applique sur des espaces discrets, tel que présentés en partie 3.1.1 (page 14).

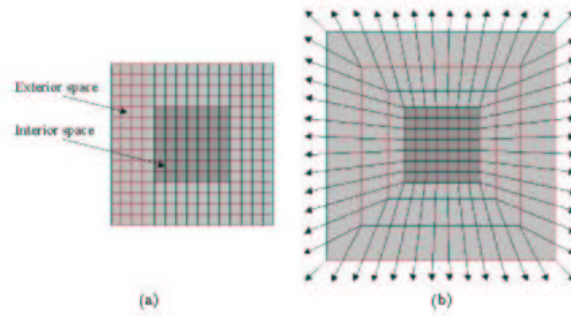


FIG. 3.3 – (a) un espace voxélique traditionnel. (b) un espace voxélique étendu. Tiré de [SMC01]

- 
1. Initialisation du volume contenant la scène en marquant tous les voxels opaques.
  2. Pour tous les voxels de la surface :
    - Tester la cohérence depuis les caméras qui sont "devant lui".
    - Si le voxel n'est pas cohérent, l'éliminer (le marquer transparent).
  3. Répéter la deuxième étape jusqu'à ce qu'il ne reste que des voxels cohérents.
- 

FIG. 3.4 – Algorithme basique du Voxel Coloring.

### Cohérence photométrique

Les algorithmes de sculpture utilisent la cohérence photométrique, introduite par [SD97], pour distinguer un point d'une surface d'un autre (étape 2 de l'algorithme 3.4 p.16). Comme expliqué en 2.4.1 (page 11), le critère de cohérence photométrique se sert de la physique des matériaux (ici lambertiens) et de l'optique pour infirmer ou non qu'il soit possible qu'un point de l'espace puisse être sur une surface. La cohérence d'un ensemble de couleur peut être évaluée par une déviation standard, ou aussi par le maximum des distances  $L_1, L_1, \dots, L_\infty$  entre chacun des couples possible dans l'espace des critères. Chacune de ces mesures peut être évaluée pour l'ensemble des critères des pixels d'un même voxel. Le voxel est accepté comme cohérent si la mesure reste en dessous d'un certain seuil. Dans le cas de la cohérence photométrique, les critères d'évaluation sont les composantes du pixel dans l'espace des couleurs.

Les scènes réelles présentent souvent des surfaces abruptes avec des frontières marquées. Certains voxels, se projetant le long de ces frontières, peuvent être évalués comme incohérents alors que situés effectivement sur la surface. [SCMS] explique que le problème peut être en partie évité en adaptant au mieux le seuil.

### Space Carving

En 1998 [KS] présente le **Space Carving**, une implémentation qui reprend l'algorithme du **voxel coloring**, en y intégrant le principe du **plane sweep** suivant les 6 directions de l'espace. Quelques voxels incohérents peuvent cependant passer outre la sculpture. Car cer-



tains voxels ne sont jamais évalués avec toutes les caméras simultanément. Pour corriger cela, [KS98] décrit des apports permettant d'obtenir une solution exacte de l'enveloppe photométrique.

En pratique, la calibration des caméras, n'est jamais parfaite. Pour palier aux erreurs de projection, [Kut00] propose d'approximation du **Space Carving**. Un voxel, ne se projetant plus en un seul pixel, mais en un disque de pixels de rayon  $r$ . Un voxel est  $r$ -cohérent si une même couleur apparaît dans tous les disques de rayon  $r$ .

### Generalized Voxel Coloring

[CMS99] a proposé une implémentation plus simple et efficace, appelée **Generalized Voxel Coloring**. Basée sur des calculs de visibilité et de cohérence exacts, elle produit, par conséquent, une reconstruction exacte de l'enveloppe photométrique. Il existe plusieurs variantes de cette implémentation, basées sur des structures de données différentes.

### Multi-hypothesis Voxel Coloring

[PEG99] propose une approche multi-hypothèses. Une hypothèse est la couleur possible d'un voxel. Chaque voxel est initialisé avec un ensemble d'hypothèses. Au fur et à mesure du déroulement du processus, le nombre d'hypothèses décroît, en même temps que les voxels sont éliminés.

La décision d'éliminer un voxel est prise avec toutes les images simultanément. C'est la différence essentielle entre cette approche et les autres. Malheureusement, la structure de données est nécessairement plus complexe et coûteuse.

## 3.2 Synthèse d'un modèle surfacique

Cette approche vise à calculer directement le polyèdre de l'enveloppe visuelle sans discrétisation volumique. Les premiers travaux sur le domaine sont dus à Baumgart [Bau74]. Il proposa une méthode de reconstruction de ce que l'on appellera plus tard enveloppe visuelle. En rétroprojetant les silhouettes des objets, on obtient des cônes généralisés. L'intersection de tous ces cônes définit l'enveloppe visuelle. La méthode s'appuie sur une approximation polygonale des silhouettes dans les images. Le cône visuel est alors défini comme un polyèdre. [BB95] propose une autre approximation, du second ordre, adaptée à la modélisation des surfaces lisses.

### 3.2.1 Surface polyédrique

Des implémentations récentes de l'approche surfacique permettent une telle reconstruction en temps-réel [MBM01] [Fra02], en tirant parti de la géométrie épipolaire et de la structure projective de l'enveloppe visuelle. Comme le montre la figure 3.5, chaque segment du contour vectorisé d'une image  $i$  définit un faisceau triangulaire dans l'espace. En outre ce faisceau est une face du cône rétroprojeté de la silhouette  $i$ . C'est pourquoi l'enveloppe visuelle comporte un certain nombre de faces qui sont contenues dans le plan de ce faisceau.

Le calcul de ces faces nécessite des intersections en 3D. L'astuce de l'algorithme consiste à effectuer ces intersections en 2D dans les plans images dès que possible et de profiter de la structure projective de l'enveloppe visuelle.

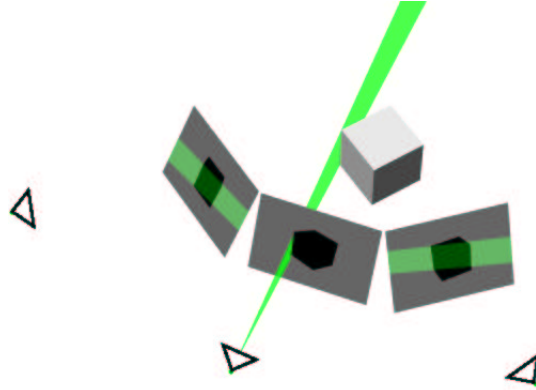


FIG. 3.5 – Faisceau triangulaire défini par un segment du contour occultant d'une image (tiré de [MBM01])

### 3.2.2 Surface du second ordre

[BB95] s'intéressent plus particulièrement à des points reconstruits par une approximation locale au second ordre de la surface. La reconstruction est effectuée d'après les contours extraits d'une séquence d'images calibrées. La surface peut ainsi être approximée à l'aide de trois contours consécutifs.

## 3.3 Synthèse d'une nouvelle vue

Dans cette partie, nous allons présenter trois approches orientées image. Par approche orientée image, il faut comprendre une approche qui synthétise une nouvelle image de la scène, d'un point de vue virtuel. Ce type d'approche présente l'avantage d'échantillonner au mieux la reconstruction pour le point de vue désiré. D'un autre côté, cela impose de calculer à nouveau la reconstruction pour chaque changement de point de vue. La première des méthodes présentée, **IBVH**, synthétise l'image sur la base de la reconstruction de l'enveloppe visuelle. Les deux suivantes présentent un processus similaire étendu cette fois à l'enveloppe photométrique.

### 3.3.1 Enveloppe visuelle par une approche image (IBVH)

[MBR<sup>+</sup>02] propose une approche image basée sur la reconstruction de l'enveloppe visuelle orientée pour une utilisation en temps-réel. La reconstruction s'effectue via l'évaluation d'intervalles de cohérence le long des lignes de vue. La figure 3.6 (page 19) illustre les 3 phases du système :

1.  $l_v$  est le rayon de projection correspondant au pixel à calculer de l'image virtuelle (en bleu).  $l_v$  est ensuite projeté en  $l_v^i$  sur les  $n$  images de référence  $I_I$ .
2. L'intersection de  $l_v^i$  avec la silhouette de l'image  $I_I$  fournit des intervalles dans lesquels l'objet n'est pas.
3. Les intervalles calculés pour chaque image de référence sont ramenés sur le rayon initial et y sont intersectés

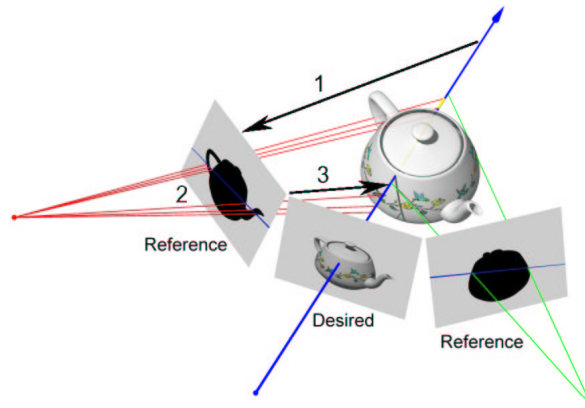


FIG. 3.6 – Enveloppe visuelle dans l'espace image - tiré de [MBR<sup>+</sup>02].

### 3.3.2 Enveloppe photométrique par une approche image

[MI02] décrit une méthode pour synthétiser une vue virtuelle sur la base d'une reconstruction de l'enveloppe photométrique. De manière similaire à **IBVH**, le rayon de projection du pixel à synthétiser est reprojété dans les vues de références (figure 3.7 (a) et (b) ). Par contre, la suite du processus diffère. Chaque droite épipolaire est redimensionnée de manière à présenter la même information. L'ensemble des droites ainsi corrigées est ensuite stocké (figure 3.7 (c) ). La question est de connaître la couleur à afficher. D'après l'hypothèse de cohérence photométrique, un point est cohérent si toutes ses projections sont cohérentes. D'autre part, pour que le point soit visible, il doit être masqué par un autre. De ces deux axiomes, [MI02] déduit que la couleur à afficher en un pixel est la première couleur présente simultanément sur l'ensemble des droites épipolaires corrigées. La méthode a deux limites majeures. Seule une image d'un point de vue situé à l'intérieur de la scène peut être synthétisée. D'autre part les problèmes de visibilité ne sont malheureusement pas décrits dans l'article.

Plus récemment, [GGS02] présente une autre approche image de l'enveloppe photométrique : **IBPH**. Plus proche de **IBVH** (section 3.3.1) que la méthode [MI02], elle tire aussi avantage de la géométrie épipolaire pour créer une reconstruction dépendante du point de vue. Les résultats de cette méthode sont plus précis que **IBVH** et [MI02], puisqu'ils vérifient les contraintes photométriques et de visibilité. L'implémentation est orientée pour le temps-réel. L'algorithme commence par calculer l'enveloppe visuelle selon **IBVH** (cf

section 3.3.1). Est ensuite évaluée la cohérence des points de l'enveloppe visuelle les plus proches, le long de chaque ligne de vue de l'image virtuelle. Si un point est éliminé, un suivant est considéré un petit peu plus loin<sup>1</sup> le long de la même ligne de vue. L'étape est répétée jusqu'à ce qu'un point cohérent soit trouvé, ou que l'enveloppe visuelle ait été traversée. Au quel cas, il n'y a pas de point cohérent le long du rayon.

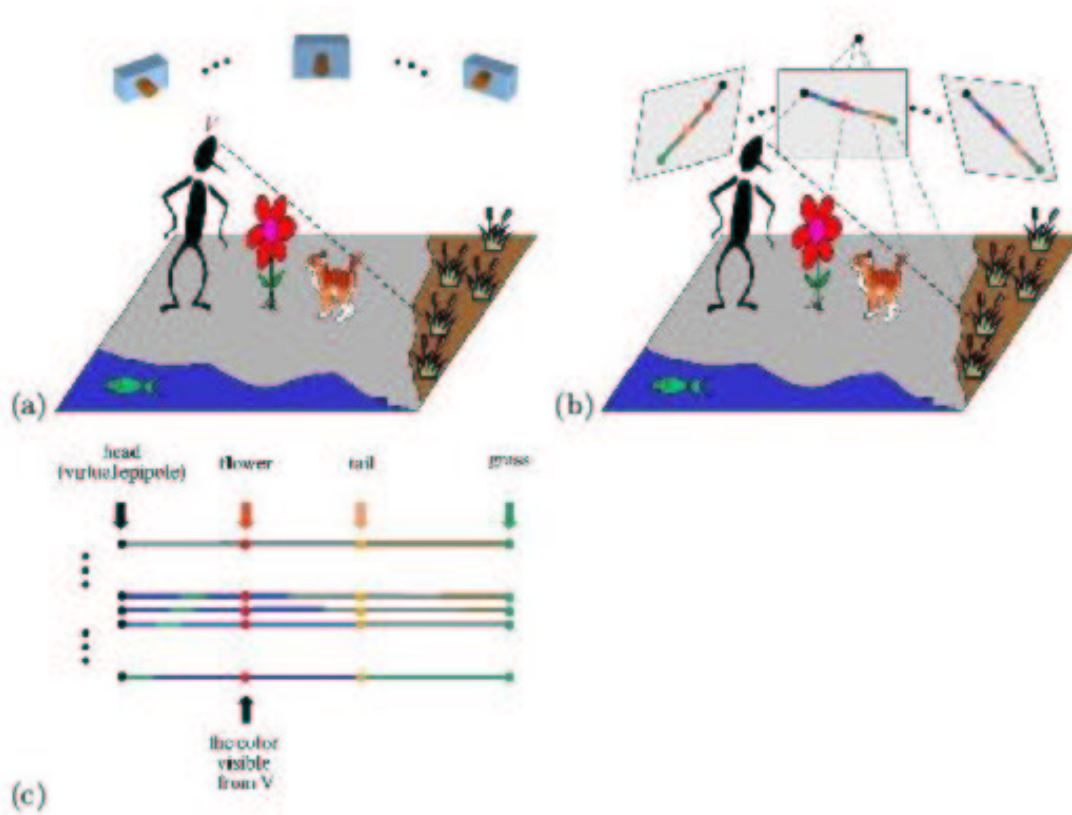


FIG. 3.7 – Enveloppe photométrique dans l'espace image (a) Le point de vue virtuel génère des rayons de projection. (b) Ces rayons sont reprojétés en droite épipolaire dans les images de références. (c) L'ensemble de droites épipolaires d'un même rayon sont redimensionnées, stockées. La couleur à afficher est la première couleur cohérente dans toutes les vues. - tiré de [MI02].

<sup>1</sup>un peu plus loin est une notion très subjective. Cela veut avant tout dire, que la construction du point se fait par une discrétisation régulière, à définir. Cette discrétisation donne lieu à des optimisations, voir [?] pour plus de précisions.

## Chapitre 4

# Extraction d'informations clefs dans l'image

Notre méthode vise à reconstruire l'enveloppe photométrique à partir d'image. L'innovation de notre méthode réside dans la création d'une discrétisation proche de l'enveloppe photométrique. Pour créer cette discrétisation, nous commençons par créer des points 3D de l'enveloppe photométrique. Ces points sont déterminés à l'aide du principe de triangulation. Pour effectuer une triangulation, il est au préalable nécessaire de mettre en correspondance des points dans les différentes images.

Dans l'absolu, nous pouvons faire une mise en correspondance de tous les pixels de chaque image avec ceux des autres images. La précision de l'espace ainsi défini étant maximale pour le jeu d'images de références. Mais la quantité de points d'une telle discrétisation est trop importante ( $nbPixel^{(nbImage)^2}$ ). En pratique, pour limiter la quantité de points nous ne considérons que des points caractéristiques de l'image. Les points les plus riches d'informations sont ceux qui se projettent entre deux zones de couleurs homogènes (voir figure 2.5 page 13). La segmentation image permet d'identifier des régions de couleurs. Les frontières de ces régions servent de point de départ pour la triangulation.

Nous présentons ci-après deux algorithmes de segmentation couleur. Nous présentons ensuite deux méthodes permettant de discrétiser le résultat de la segmentation, afin de la rendre exploitable par les outils de triangulation conventionnels.

### 4.1 Algorithmes de segmentation image

#### 4.1.1 Blobworld [CTB<sup>+</sup>99]

Blobworld est un outil de segmentation automatique d'images couleurs. La segmentation délimite grossièrement les objets ou parties d'objet contenues dans l'image. "Blobworld" propose une représentation de chaque région par ces caractéristiques de couleur, texture, et géométrie. Cet outil de segmentation est orienté pour chercher des images dans un base de données.



FIG. 4.1 – Blobworld. Tiré du site <http://elib.cs.berkeley.edu/photos/blobworld/>

Cette segmentation à l'avantage de mettre en évidence les contours des objets. Cependant, elle crée des régions qui peuvent être caractérisées par de des textures prononcées, comme le pelage du tigre dans la figure 4.1. Ce sera pour nous une segmentation trop grossière, qui ne fournit pas assez de régions.

#### 4.1.2 Edgeflow

[MM97] présente une nouvelle approche de segmentation couleur automatique, appelée JSEG. Cette approche permet d'identifier des régions, même texturées, dans une image. L'approche assure que :

- chaque région de l'image contient une répartition uniforme de motif texturé.
- l'information de chaque région peut être résumée par quelques couleurs choisies. Ce qui est vrai pour les scènes réelles.
- les couleurs de deux régions voisines sont distinguables. Ce qui doit être vrai pour toute segmentation.

Les deux étapes indépendantes de la segmentation sont :

1. Quantification des couleurs. Cela consiste à chercher quelques couleurs représentatives de l'image, puis à associer chaque pixel à l'une des classes (couleurs caractéristiques). Cet étiquetage crée une carte des classes.
2. Segmentation spatiale



FIG. 4.2 – Segmentation "edge flow" couleur et avec des textures. Tiré du site <http://vision.ece.ucsb.edu/segmentation/edgeflow/>

L'implémentation est paramétrable. Ce qui nous permet d'obtenir le niveau de précision souhaité. Malheureusement, la nécessité de fournir des paramètres limite l'automatisation de la segmentation.

## 4.2 Discrétisation des contours

Pour pouvoir exploiter l'information de contour, il est préférable de la discrétiser. La discrétisation du contour se fait par une polygonalisation. Nous présentons une méthode « naïve » (figure 4.3 (a)), puis une méthode optimale (figure 4.3 (b)).

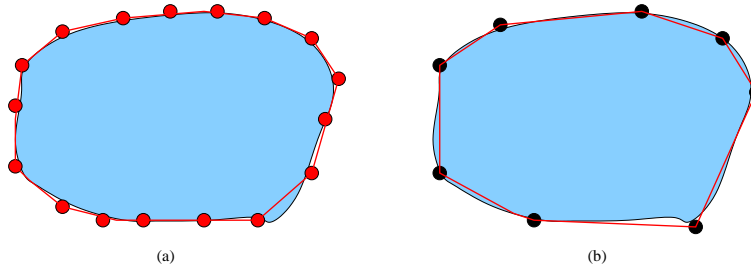


FIG. 4.3 – (a) polygonalisation naïve. (b) polygonalisation optimale.

### 4.2.1 Polygonalisation naïve

Comme l'indique le titre de cette section, la polygonalisation naïve est naïve. Les contours sont parcourus, et des points sont créés à intervalles réguliers, reliés par des segments. Attention, petite difficulté, pour les contours topologiquement circulaires, il faut créer un segment qui ferme le polygone.

### 4.2.2 Polygonalisation optimisée

L'algorithme présenté est incrémental. Il crée le nombre de segments nécessaires pour que la distance de tous les points du contour aux segments soit inférieure à un seuil défini.

---

**tant que**  $\exists (p_i \wedge s_n) \wedge \neg \exists (p_j \wedge s_m)$  **tel que**  $(dist(p_i, s_n) < seuil) \wedge (dist(p_i, s_n) < dist(p_j, s_m))$   
 scinder  $s_n$  par  $p_i$

---

FIG. 4.4 – Algorithme de polygonalisation optimisé - tiré de [Sch03].

Où  $p_i$  et  $p_j$  sont deux points différents du contour,  $s_n$  et  $s_m$  deux segments (éventuellement deux fois le même), et  $dist(p, s)$  la distance d'un point à un segment.

## Chapitre 5

# Calcul de points sur la surface de l'enveloppe photométrique

Le but de notre méthode est de reconstruire l'enveloppe photométrique via une discrétisation adaptée. Les régions et leurs contours, associés aux contraintes épipolaires, permettent de trianguler des points 3D situés sur l'enveloppe. Dans ce chapitre, nous allons expliquer comment trianguler ces points à partir des points et segments des images.

Considérons un point 3D à reconstruire. Ce point se projette entre 2 régions dans au moins 2 images. Nous allons donc reconstruire tous les points vérifiant cette propriété. Nous procédons en deux étapes. Premièrement, nous construisons des points 2D situés sur une frontière dans deux images et satisfaisant les contraintes épipolaires. Deuxièmement, Nous triangulons ces points images afin d'obtenir les points 3D désirés.

### 5.1 Création des points sur le contour

Trianguler un point 3D ne peut pas se faire avec deux points 2D quelconques. Les points 2D doivent satisfaire la contrainte épipolaire. Les points obtenus par la discrétisation des contours ne satisfont pas forcément cette contrainte. Nous allons donc en calculer. Nous traçons la droite épipolaire (figure 5.1 (1) ). Cette droite coupe en  $p_j$  les contours (figure 5.1 (2) ). Nous calculons  $p_j$  de la manière suivante :

$$p_j = el_j \wedge sl_j \quad (5.1)$$

$$el_j = F_{ij} \cdot p_i \quad (5.2)$$

$$sl_j = s1_j \wedge s2_j \quad (5.3)$$

Où :

- $p_j$  est le point 2D du contour dans l'image  $I_j$ , construit tel qu'il vérifie la contrainte épipolaire avec le point  $p_i$ .
- $sl_j$  est la droite passant par les extrémités  $s1_j$  et  $s2_j$  du segment du contour à intersecter dans l'image  $I_j$ .
- $el_j$  est la droite épipolaire correspondant au point  $p_i$  dans l'image  $I_j$ .



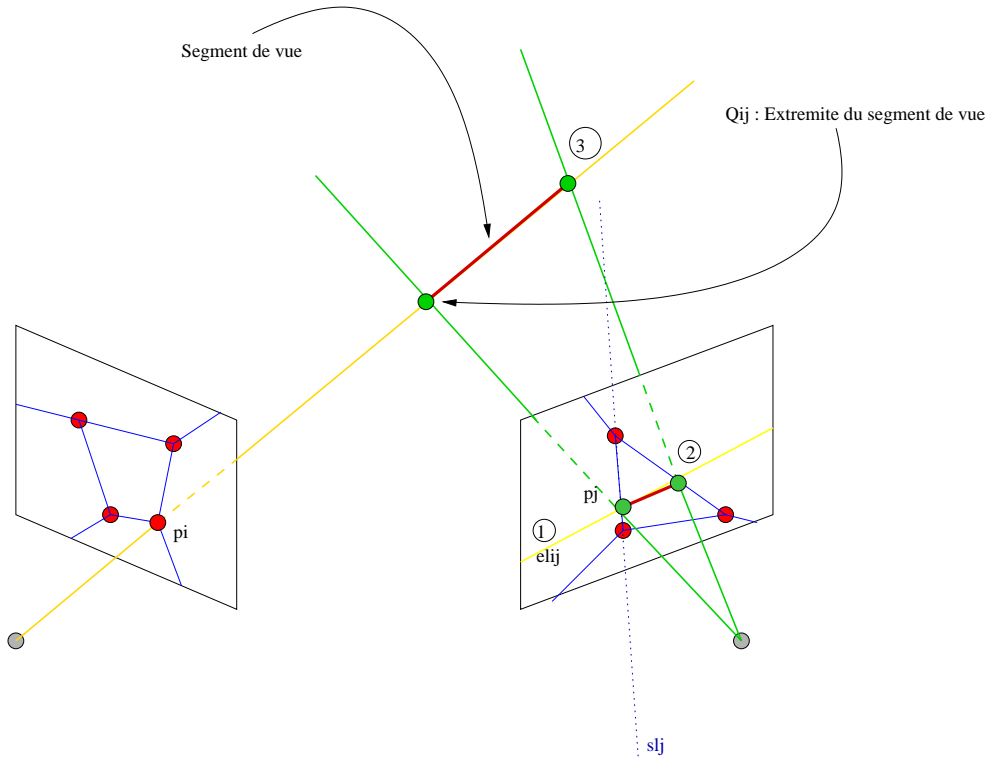


FIG. 5.1 – Triangulation de points.

## 5.2 Création des segments épipolaires

Les points 2D  $p_j$  calculés précédemment reprojetés sur la ligne de vue  $l_i$  du point  $p_i$  délimitent des intervalles, appelés segments de vue (figure 5.1 (3)). Chaque segment de vue correspond à la région dont il est issu.

$$Q_{ij} = l_i \wedge l_j \quad (5.4)$$

La ligne de vue a pour expression :

$$l_i = p_{i\infty} \wedge c_i \quad (5.5)$$

Où :

–  $p_{i\infty}$  le point à l'infini le long du rayon de projection de  $p_i$  d'expression :

$$p_{i\infty} = \begin{pmatrix} \overline{M}_i^{-1} \cdot p_i \\ 0 \end{pmatrix}$$

–  $c_i$  le centre optique de l'image  $I_i$  d'expression :

$$c_i = \begin{pmatrix} t_i \\ 1 \end{pmatrix}$$

–  $p_i$  et  $p_j$ , les points 2D servant à trianguler (voir figure 5.1).

Tous les points se projetant en dehors d'une silhouette ne serviront pas à la reconstruction, puisque en dehors de l'enveloppe visuelle et photométrique. Pour économiser une perte de temps inutile, ils peuvent être éliminés d'office.

Dans notre implémentation, nous avons repris l'optimisation décrite sous le nom de **edge-bins** dans [MBM01]. Cette optimisation accélère la recherche de segments dans une image, en les triant par leur angle par rapport à l'épipole.

## Chapitre 6

# Discrétisation volumique

Grâce à l'étape décrite précédemment, nous avons construit des points dont certains sont sur l'enveloppe photométrique (partie 5 page 24). Ces points ne fournissent ni un modèle surfacique ni volumique. D'autre part, la majorité de ces points n'ont pas de sens physique. Nous avons décidé d'affronter le problème par une méthode de sculpture, de type **voxel coloring**. Nous avons donc besoin de discrétiser l'espace en volumes élémentaires, tout en conservant le bénéfice du calcul des points. Une triangulation nous permettrait de subdiviser le volume de travail en tétraèdres, suivant les points reconstruits, à un moindre coût.

### 6.1 Triangulation de Delaunay

La triangulation de Delaunay possède la propriété suivante : la sphère circonscrite à chaque tétraèdre ne contient aucun autre point de la triangulation. Cette triangulation a aussi la propriété de minimiser la différence entre les angles des tétraèdres.

Avec la triangulation de Delaunay, il est possible que les facettes des tétraèdres ne suivent pas à la surface de l'objet. En effet, rien ne garantit que le découpage suive la logique du découpage des régions.

En pratique, cela pourra poser quelques problèmes, mais généralement la taille des tétraèdres est suffisamment petite pour négliger cet aspect.

Dans l'implémentation, nous avons utilisé la bibliothèque QHull en langage C. [EBE] déclare : "Utilisé QHull n'est pas souhaitable pour la génération de maillage ou la triangulation de surfaces arbitraires."

### 6.2 Les limites de la triangulation

La triangulation de Delaunay classique ne fournit pas forcément des tétraèdres dont les faces suivent les contours de l'objet (voir figure 6.1 page 28). Nous avons constaté que ce défaut n'entrave pas trop la reconstruction. Toutefois, cela dégrade la reconstruction par irrégularités sur la surface des objets reconstruits.

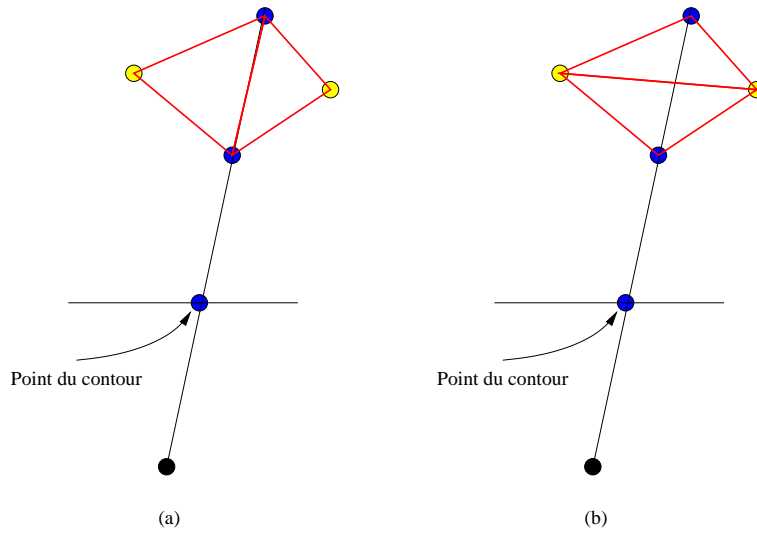


FIG. 6.1 – Plusieurs triangulations sont possibles : (a) triangulation qui suit le contour de l'objet, (b) La triangulation coupe l'objet.

# Chapitre 7

## Sculpture

### 7.1 Approche

L'ensemble des tétraèdres de la triangulation de Delaunay ne correspond pas à l'enveloppe photométrique, ni même à l'enveloppe visuelle. La triangulation fournit une enveloppe convexe autour de points. Il faut donc éliminer des tétraèdres. Dans le chapitre 3 (page 14) nous avons vu comment les algorithmes de type **voxel coloring** sculptent les voxels jusqu'à obtenir l'enveloppe photométrique. Dans cette partie, nous allons exposer les différentes approches que nous avons expérimentées pour généraliser la technique de sculpture à notre ensemble de tétraèdres. Le principe du **voxel coloring** est de parcourir les voxels les uns après les autres et d'éliminer ceux qui ne sont pas cohérents. Il existe des critères qui permettent de décider de la cohérence. Ce qui, au départ est un volume brut, devient l'enveloppe photométrique de l'objet. C'est en cela que la méthode s'apparente à de la sculpture. La forme cubique du voxel, permet de faire des simplifications que le tétraèdre ne permet pas.

Nous allons donc étudier les trois points essentiels de la méthode, et nous verrons quelles adaptations furent nécessaires.

Le premier point important est de bien choisir le critère de cohérence photométrique. Ce critère est une mesure qui permet de décider si un tétraèdre doit être éliminé ou pas. Les tétraèdres n'apportent ici aucune complication. Nous verrons deux méthodes existantes satisfaisantes.

Le second point clef est d'établir un test de visibilité. Il est nécessaire de savoir si un tétraèdre et les faces associées sont visibles depuis un point de vue, notamment pour évaluer le critère de cohérence photométrique. Dans le cas du **voxel coloring**, il existe une multitude d'algorithmes, allant d'une rapide approximation à un calcul plus long et plus précis. Le problème se simplifie avec les tétraèdres. La géométrie algorithmique fournit une solution optimale. Nous reprendrons ici l'algorithme, avec les détails liés à notre problème.

Le troisième point critique est l'ordre de parcours des tétraèdres. En effet, celui-ci influence directement la complexité du problème.

Finalement nous exposerons la méthode de sculpture dans son ensemble.

## 7.2 Cohérence photométrique

Les critères de cohérence photométrique servent à décider de la réalité physique d'un volume unité (ici, le tétraèdre). Pour simplifier le problème, nous approximerons le tétraèdre par un ensemble de points. Evaluer la cohérence d'un tétraèdre se fera par l'évaluation de la cohérence des points l'approximant.

Le point sera photométriquement cohérent si les couleurs des pixels où il se projette sont cohérentes. Nous ne considérons évidemment que les pixels où le point est visible. Par couleurs cohérentes, nous voulons dire que les couleurs vérifient toutes les propriétés physiques du point. Les caractéristiques d'une surface ne se limitent pas seulement à la couleur. La spécularité, la réflexion et la transparence sont des facteurs qui entrent en compte. Pour ramener le problème à une complexité acceptable, hypothèse est prise que la scène ne contient que des objets lambertiens. Il est fait abstraction de toutes les informations autres que la couleurs. Le problème se simplifie en ces termes : la projection d'un même point de l'espace donnera toujours la même couleur, indépendamment du point de vue. L'hypothèse est forte, mais tolérable dans la plus part des scènes réelles.

### 7.2.1 Approximation du tétraèdre

Evaluer la cohérence sur l'ensemble de la surface d'un tétraèdre n'est pas forcément nécessaire. Un ensemble de points fournit une approximation acceptable dans la plupart des cas. Pour évaluer la cohérence d'un tétraèdre, nous nous contenterons d'évaluer les quatre points des centres des quatre faces du tétraèdres.

### 7.2.2 La mesure de distance

L'ensemble des projections d'un point dans les différentes images fournit un ensemble de couleurs. Théoriquement, si tous les éléments de cet ensemble représentent la même couleur, tous ces éléments doivent être égaux entre eux. En pratique, le signal de l'image est bruité. Pour éliminer une partie du bruit, nous travaillons dans l'espace de couleur  $RG_{normalise}$  décrit en section 2.4.2 (page 11). Pour limiter la sensibilité au bruit sur les deux composantes restantes, nous utilisons une mesure de distance, associée à un seuil de tolérance. La mesure doit permettre de différencier un ensemble de couleurs dont une seule est différentes des autres (7.1 (b) 30), d'un ensemble l'ensemble de couleurs assez proches. (7.1 (a) 30).

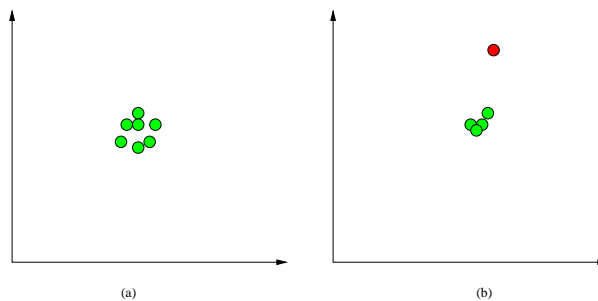


FIG. 7.1 – (a) Ensemble cohérent de couleurs. (b) Ensemble incohérent de couleurs.

### Variance

Si nous considérons  $\Sigma$  la matrice de covariance de la distribution de la couleur, son déterminant est un scalaire représentant la variance de la distribution.

La matrice de covariance  $\Sigma$  du pixel  $x$ , dont voici l'expression :

$$\Sigma_x = \begin{pmatrix} var(x_1) & cov(x_1, x_2) \\ cov(x_1, x_2) & var(x_2) \end{pmatrix} \quad (7.1)$$

$$det(\Sigma_x) = var(x_1) \times var(x_2) - cov(x_1, x_2) \times cov(x_1, x_2) \quad (7.2)$$

où

- $var(x_i)$  avec  $i \in \{1, 2\}$  est la variance sur l'axe  $i$  de l'espace de couleurs.
- $cov(x_i, x_j)$  avec  $i < j \in \{1, 2\}$  est la covariance entre les axes  $i$  et  $j$  de l'espace de couleurs.

La matrice de covariance  $\Sigma$  peut être calculée de la manière suivante :

$$\Sigma_x = \frac{1}{M_1} \times M_3 - \frac{M_2 \times M_2^T}{M_1^2} \quad (7.3)$$

avec :

$$M_1 = \sum_{0 \leq n \leq N-1} 1 \quad (7.4)$$

$$M_2 = \sum_{0 \leq n \leq N-1} x_n \quad (7.5)$$

$$M_3 = \sum_{0 \leq n \leq N-1} x_n \times x_n^T \quad (7.6)$$

où

- $M_1$  est un scalaire
- $M_2$  est un vecteur de taille égale au nombre de dimensions de l'espace
- $M_3$  est une matrice  $2 \times 2$

### Distance maximum à la moyenne

La variance ne permet malheureusement pas toujours de différencier les 2 cas de la figure 7.1 (page 30). L'importance d'un point est pondéré par le nombre d'échantillons. Pour pallier à cela, nous prenons comme mesure le maximum des distances de chaque couleur à la moyenne.

## 7.3 Test de visibilité

### 7.3.1 Visibilité d'un tétraèdre

Une image contribue à la mesure de cohérence d'un tétraèdre seulement si celui-ci y est visible. Nous approximerons la visibilité d'un tétraèdre par la visibilité de son centre. Nous nous ramenons ainsi à un problème de géométrie algorithmique connu :

---

```

Tetraedre tetraCourant := tetraedre a tester
Booleen visible := vrai
Ligne L1 := ligne de vue de la camera au tetraedre a tester
Tant que visible et tetraCourant != tetraedre exterieur
  facette1 := facette intersectant la L1 la plus proche
  tetraCourant := voisin de tetraCourant par face facette1
  visible := tetraCourant non coherent

```

---

FIG. 7.2 – Algorithme de parcours des tétraèdres.

De manière intuitive (cf. fig 7.3), l'algorithme 7.2 parcourt la triangulation du tétraèdre à tester jusqu'à celui contenant la caméra. Si sur le parcours, un seul tétraèdre est cohérent, alors il le masque. Il n'est donc pas visible.

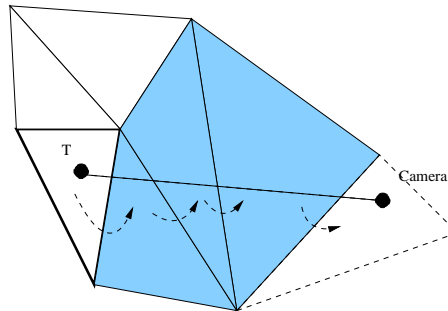


FIG. 7.3 – Parcours de la triangulation, pour évaluer la visibilité.

### 7.3.2 Visibilité d'une face d'un tétraèdre

La face d'un tétraèdre n'est visible que si le tétraèdre lui-même l'est aussi. Une solution ayant déjà été proposée dans la sous-section précédente, nous n'expliquerons pas à nouveau ce test. Prenons pour acquis que le tétraèdre soit visible. Une face est visible si et seulement si, des deux faces coupées par la ligne de vue, elle est la plus proche de la caméra. Le cas limite se présente pour les tétraèdres plats. Les approximations numériques peuvent fausser le test. Dans ce cas, toutes les faces du tétraèdres sont déclarées visibles.

## 7.4 Ordre de parcours des tétraèdres

La cohérence d'un tétraèdre dépend de celle des autres tétraèdres. Cette récursion est liée au problème de visibilité. En effet, la cohérence ne doit être évaluée qu'avec les vues desquelles le point est visible. La visibilité d'un tétraèdre dépend de la cohérence de ceux situés devant lui (cf. 7.3). Chaque fois qu'un tétraèdre est éliminé, la visibilité de ceux qui se trouvent derrière lui est changée. A chaque suppression d'un tétraèdre, il faut évaluer à nouveau la cohérence de ceux qu'il masquait. Pour permettre d'évaluer les voxels en une



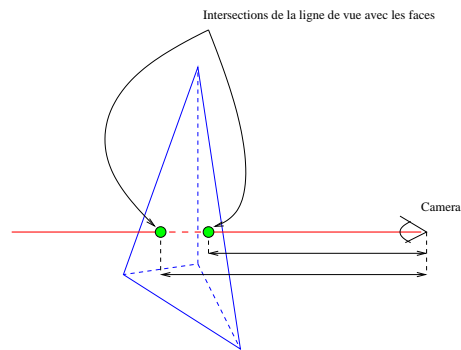


FIG. 7.4 – Visibilité d'une face.

seule passe, *Seitz et Dyer* imposent ce qu'ils appellent une *contrainte d'ordre de visibilité* sur la position des caméras : par exemple, en plaçant toutes les caméras du même côté de la scène. [KS98] développa la technique de **plane sweep** pour permettre un placement arbitraire des caméras.

Il n'existe pas à notre connaissance de méthodes qui permettent de sculpter les tétraèdres en un nombre d'itérations défini. Nous proposons ici deux méthodes qui limitent en pratique le nombre d'itérations, par rapport à un parcours aléatoire.

#### 7.4.1 Parcours via le *plane sweep*

##### Définition

Le **plane sweep** est un artefact décrit dans [SD99] qui permet deux simplifications :

1. un test de visibilité accéléré et simplifié.
2. une convergence de l'algorithme bien plus rapide.

Le **plane sweep** est un procédé proposé [KS98] servant à guider l'ordre de parcours des voxels. La grille voxelique est examinée plan après plan, suivant les 6 directions de l'espace (selon les 3 axes principaux, dans le sens positif et négatif). Seules les caméras situées en avant du plan sont consultées pour évaluer la cohérence. Nous sommes ainsi ramené au cas où toutes les caméras sont situées d'un même côté de la scène. Le nombre d'itérations nécessaires en est diminué, puisqu'il faut 6 passes pour évaluer toute la scène.

##### Appliqué aux tétraèdres

Le parcours selon les **plane sweep** ne garantit pas que les tétraèdres occultants seront traités avant les occultés. Comme l'illustre la figure 7.5, l'ordre de parcours peut être différent pour toutes les caméras. Il faut donc itérer le processus jusqu'à ce que la quantité de tétraèdres éliminés soit nulle (ou non significative). Cependant, la technique de **plane sweep** permet une convergence acceptable (voir le graphique de convergence 8.4 page 38).

#### 7.4.2 Liste des tétraèdres de surface

Nous avons pensé à une autre alternative pour accélérer la convergence de la sculpture. L'idée de base est qu'il n'est pas nécessaire de parcourir toute la liste des tétraèdres. Seuls

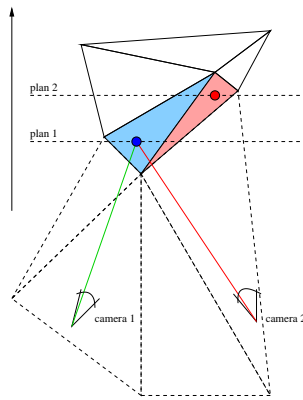


FIG. 7.5 – Les limites du plane sweep : La caméra 1 voit en premier le tétraèdre bleu clair, alors que la caméra 2 voit le rose.

ceux situés en surface sont susceptibles d'être éliminés. En effet, pour que la cohérence puisse être évaluée, le tétraèdre doit être visible d'au moins 2 caméras. S'il n'est pas en surface, il n'est visible d'aucune caméra.

Il suffit de tenir à jour une liste des tétraèdres de surface.

Pour initialiser la liste, il faut parcourir tous les tétraèdres et ajouter à la liste ceux qui ont un voisin incohérent (la scène est bordée de tétraèdres incohérents).

Pour maintenir la liste à jour, il faut ajouter à la liste tous les voisins d'un tétraèdre à éliminer, sauf s'ils sont déjà présents dans la liste ou bien déjà éliminés.

## 7.5 Sculpture dans la triangulation

Nous avons implémenté l'algorithme 7.6 (page 35).

---

```

pour chaque direction  $d$  faire
  pour chaque tetraedre  $t$  suivant la direction  $d$  faire
    si le  $t$  est visible et n'a pas ete elimine alors
      pour chaque face  $f$  de  $t$  faire
        reel  $moyenne$  pour chaque camera  $c$  faire
          si  $f$  est visible de  $c$  alors
            mettre a jour les caracteristiques photometriques de  $f$ 
            avec la contribution du pixel de la projection du centre de  $f$  dans  $c$ 

          si moins  $f$  visible de moins d'une camera alors
            arreter

        entier  $max := 0$ 
        pour chaque camera  $c$  faire
          si  $f$  est visible de  $c$  alors
             $max := \max(max, \text{dist}(moyenne, \text{couleur du centre de } f \text{ dans } c))$ 

        si  $max > \text{seuil}$  alors
          eliminer le  $t$ 

```

---

FIG. 7.6 – Algorithme de sculpture des tétraèdres.

# Chapitre 8

## Résultats

Les résultats présentés ci-après ont été réalisés à partir de 12 images synthétiques similaires à celle de la figure 8.1, d'un cube contenant une concavité.

La segmentation a été effectuée par le programme **JSEG** (cf partie 4.1.2 22). Les régions ont été polygonalisées par la méthode naïve.

L'étape de sculpture décrite en 7.6 à été répétée jusqu'à ce que le nombre de tétraèdres éliminés ne soit plus significatif (5 fois).

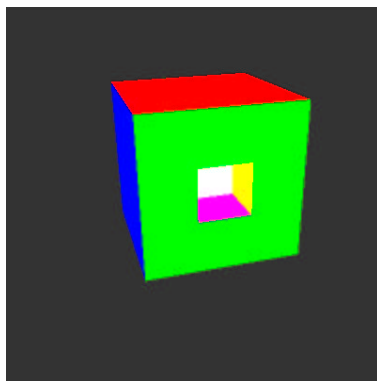


FIG. 8.1 – Image servant à la reconstruction d'un cube avec une concavité.

### 8.1 Apport de la photométrie

Dans l'implémentation nous avons séparé l'étape de reconstruction de l'enveloppe visuelle et de celle de la reconstruction de l'enveloppe photométrique. Sur la figure 8.2, nous pouvons aisément constater l'apport des contraintes photométriques. Sur l'image de droite, la zone de concavité est bien marquée. La légère pointe à l'intérieur de la concavité est caractéristique l'enveloppe photométrique (similaire à la figure 2.4 page 13).

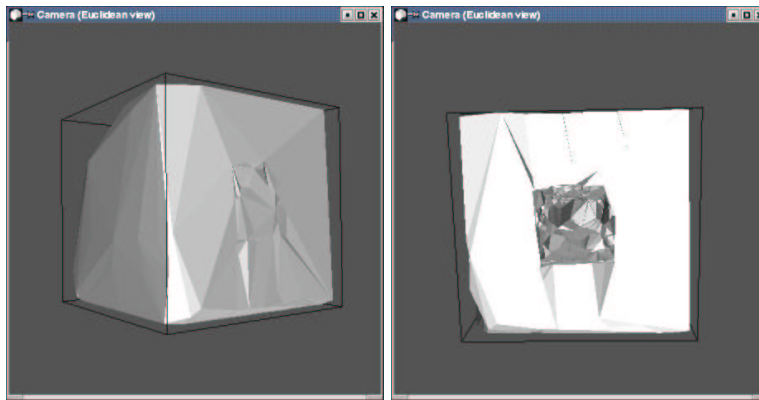


FIG. 8.2 – A gauche : résultat de l’enveloppe visuelle. A droite : résultat de l’enveloppe photométrique.

## 8.2 Apport de la discrétisation adaptée

L’étape de construction de points pertinents (partie 5 page 24) doit générer une discrétisation suivant au mieux le relief de l’objet. Dans la figure 8.3 (page 37), nous pouvons comparer avec les autres types de discrétisation. Nous pouvons aisément constater deux choses :

- dans les trois cas, la concavité du cube est bien reconstruite.
- la discrétisation calculée via la segmentation image fournit un résultat meilleur, non seulement en terme de précision, et de fiabilité.

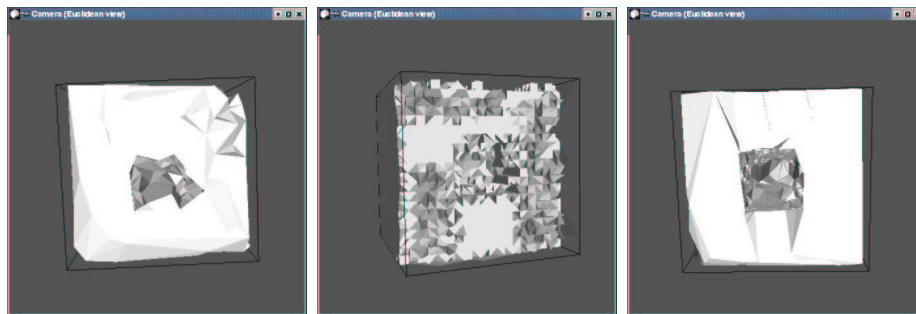


FIG. 8.3 – De gauche à droite : Reconstruction sur la base d’un espace discrétisé aléatoirement, régulièrement, ou sur la base de la segmentation image.

## 8.3 Complexité

Dans cet exemple précis, il a été nécessaire d’évaluer 9 fois tous les tétraèdres restants (voir figure 8.4 page 38) suivant les **plane sweep** dans les 6 directions de l’espace pour ne plus avoir aucun tétraèdre à supprimer.

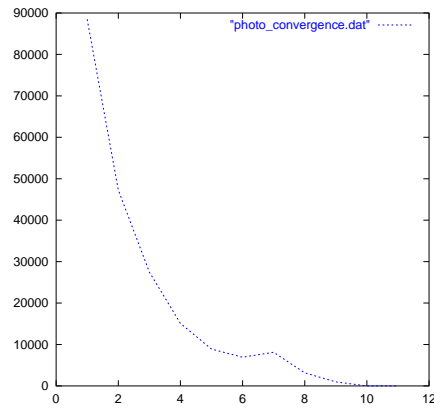


FIG. 8.4 – Convergence de la sculpture.

### 8.3.1 Triangulation imparfaite

Dans l'algorithme, tel qu'il est implémenté, les faces des tétraèdres sont approximées par leur centre. Certains tétraèdres, comme ceux de la surface de la triangulation de Delaunay, sont de grande taille. L'approximation devient alors non significative. Sur la figure 8.5 (page 39) le côté de la théière, le centre des faces est cohérent, alors que l'intégralité de la face ne l'est pas forcément.

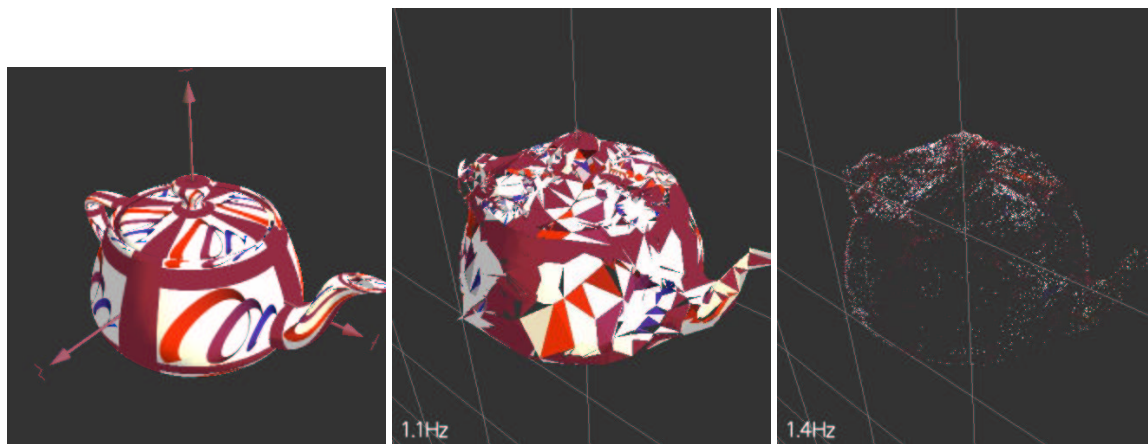


FIG. 8.5 – De gauche à droite : la vue servant à la reconstruction. La reconstruction du même point avec les tétraèdres, ou seulement les points servant à évaluer la cohérence.

## Chapitre 9

# Conclusion et perspectives

Nous avons mis au point une nouvelle méthode de reconstruction volumique. Cette reconstruction est effectuée dans un espace discret. Contrairement aux autres méthodes de **voxel coloring**, la discrétisation de l'espace est calculée au mieux pour reconstruire l'enveloppe photométrique avec plus de précision et pour un coût raisonnable. Une telle discrétisation permet d'évacuer les inconvénients liés à la discrétisation arbitraire.

Les premiers résultats sur des données synthétiques ont permis de valider notre approche. L'enveloppe photométrique est effectivement reconstruite. Cependant, les reconstructions obtenues souffrent de quelques défauts. Plusieurs des étapes implémentées doivent être améliorées pour éventuellement traiter des données réelles.

### 9.1 Améliorer les différentes étapes

L'implémentation actuelle a été pensée pour expérimenter et valider la pertinence de la méthode. Certaines étapes de l'algorithme n'ont pas été implémentées au mieux. Nous pouvons expérimenter d'autres approches pour les étapes suivantes :

- La segmentation
- Le parcours des tétraèdres pourrait être plus optimal (partie 7.4.2 page 33).
- Implémenter et tester une triangulation de Delaunay contrainte.

### 9.2 Une meilleure approximation des faces

Dans l'implémentation actuelle, les faces des tétraèdres sont approximées par leur centre. Certains tétraèdres, comme ceux de la surface de la triangulation de Delaunay, sont de grande taille. L'approximation devient alors non significative. Sur la figure 8.5 (page 39) le côté de la théière, le centre des faces est cohérent, alors que l'intégralité de la face ne l'est pas forcément.

Prendre plus de points répartis sur la face et évaluer la cohérence en chacun d'eux permettrait de rendre la reconstruction plus précise. Il reste à essayer et à explorer les différentes façons de choisir les points (aléatoirement, régulièrement, ...).



### 9.3 Diminuer le nombre de points 3D initiaux

Dans la partie 5.1 (page 24) nous expliquons comment nous reconstruisons des points 3D à partir de la segmentation image. De points situés entre deux régions dans deux images, nous définissons un point 3D. Ce point a la propriété de se projeter sur une frontière de deux régions dans plusieurs images. Le but de cette manœuvre est de reconstruire des points dont certains sont ceux qui caractérisent l'enveloppe photométrique.

Le fait que les points intéressants soient sur l'enveloppe photométrique apporte une contrainte supplémentaire que nous n'avons pas exploitée. En effet, si un point est réellement sur l'enveloppe photométrique, il est à la frontière de deux régions 3D. En 2d, il se projette donc au moins deux fois (dans des images différentes) au bord de la même région. Nous pouvons donc identifier dès la triangulation des points qui ne sont pas sur l'enveloppe photométrique.

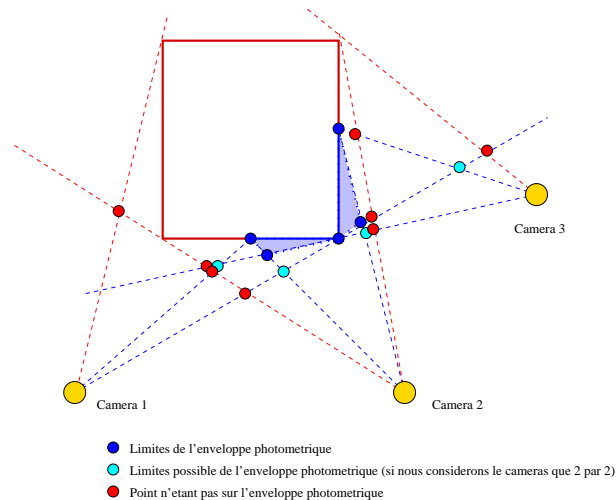


FIG. 9.1 – Certains points 3D reconstruits ne peuvent pas être sur l'enveloppe photométrique : les points n'ayant aucune frontière semblable dans l'ensemble des images.

### 9.4 Application à des images réelles

Le but ultime est de valider l'approche avec des prises de vues réelles. D'autres problèmes sont engendrés par l'application à des photos. La calibration est souvent imprécise et les images bruitées : des paramètres qui influencent la robustesse, voire la convergence de la méthode ( 8.4 page 38).

# Bibliographie

- [Bau74] Bruce G. Baumgart. *Geometric Modeling for Computer Vision*. Artificial Intelligence Project Memo AIM-249 (CS-TR-74-463), Computer Science Department, Stanford University, Palo Alto, CA, 1974.
- [BB95] E. Boyer and M. Berger. 3d surface reconstruction using occluding contours. In *Computer Analysis of Images and Patterns*, pages 198–205, 1995.
- [BF03] E. Boyer and J-S. Franco. A hybrid approach for computing visual hulls of complex objects, 2003.
- [CMS99] W. B. Culbertson, T. Malzbender, and G. G. Slabaugh. Generalized voxel coloring. In *Workshop on Vision Algorithms*, pages 100–115, 1999.
- [CTB<sup>+</sup>99] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld : A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.
- [Dye01] C. R. Dyer. Volumetric scene reconstruction from multiple views. In L. S. Davis, editor, *Foundations of Image Understanding*, pages 469–489. Kluwer, 2001.
- [EBE] Ravalet S. Renoult L. El Brichi E.H., Lacherade S. Reconstruction 3d d'une surface à partir d'un semi de points.
- [Fra02] J-S. Franco. Reconstruction 3d en temps-réel pour la réalité virtuelle. Mémoire de dea ivr, tuteur Edmond Boyer, INRIA-GRAVIR équipe iMagis, 2002.
- [GGS02] M. C. Hans G. G. Slabaugh, R. W. Schafer. Image-based photo hulls. pages 1–15. HP Laboratories Palo Alto, 2002.
- [KS] K. N. Kutulakos and S. M. Seitz. What do photographs tell us about 3d shape ?
- [KS98] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. Technical Report TR692, 1998.
- [Kut00] K. N. Kutulakos. Approximate n-view stereo. In *ECCV(1)*, pages 67–83, 2000.
- [Lau94] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2) :150–162, 1994.
- [LBP01] S. Lazebnik, E. Boyer, and J. Ponce. On how to compute exact visual hulls of object bounded by smooth surfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*. IEEE Computer Society Press, Dec 2001.

- [MBM01] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for Real-Time rendering. In *Siggraph 2001*, pages 115–126, 2001.
- [MBR<sup>+</sup>02] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2002.
- [MI02] P. Anandan M. Irani, T. Hassner. What does the scene look like from a scene point ? pages 883–897, 2002.
- [MM97] W. Ma and B. Manjunath. Edge flow : a framework of boundary detection and image segmentation, 1997.
- [PEG99] E. Steinbach P. Eisert and B. Girod. Multi-hypothesis volumetric reconstruction of 3-d objects from multiple calibrated camera views. pages 3509–3512, 1999.
- [Sch03] C. Schmidt. Note de cours de dea ivr, 2003.
- [SCMS] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. pages 81–100.
- [SD97] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring, 1997.
- [SD99] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2) :151–173, 1999.
- [SK] H. Saito and T. Kanade. Shape reconstruction in projective grid space from large number of images. pages 49–54.
- [SMC01] G. G. Slabaugh, T. Malzbender, and W. B. Culbertson. Volumetric warping for voxel coloring on an infinite domain. *Lecture Notes in Computer Science*, 2018 :109–??, 2001.
- [SS99] G. Slabaugh and R. W. Schafer. Multiresolution space carving. *International Journal of Computer Vision*, 35(2) :151–173, 1999.
- [Stu03] P. Sturm. Note de cours de dea ivr, 2003.