

Une approche hybride pour calculer l'enveloppe visuelle d'objets complexes

Jean-Sébastien Franco, Edmond Boyer

▶ **To cite this version:**

Jean-Sébastien Franco, Edmond Boyer. Une approche hybride pour calculer l'enveloppe visuelle d'objets complexes. Actes des Journées ORASIS, May 2003, Gerardmer, France. pp.67–74. inria-00349086

HAL Id: inria-00349086
https://inria.hal.science/inria-00349086

Submitted on 23 Dec 2008

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

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

Une Approche Hybride pour Calculer l'Enveloppe Visuelle d'Objets Complexes

A Hybrid Approach for Computing Visual Hulls of Complex Objects

Jean-Sébastien FRANCO Edmond BOYER

GRAVIR, Equipe MOVI, INRIA Rhône-Alpes

ZIRST Montbonnot, 655, av. de l'Europe, 38334, St-Ismier Cedex, FRANCE
Prenom.Nom@inrialpes.fr

Résumé

Nous traitons de la reconstruction de l'enveloppe visuelle à partir des silhouettes de multiples images. Nous proposons une approche hybride qui tire avantage des approches surfaciques et fournit à complexité équivalente une précision beaucoup plus importante que les approches volumétriques. Celles-ci se basent en effet sur une discrétisation de l'espace sous la forme d'une grille régulière, dont beaucoup de cellules n'apportent que peu d'information. C'est pourquoi nous introduisons une discrétisation irrégulière, basée sur une grille constituée de points échantillonnés sur la surface de l'enveloppe visuelle, et de cellules tétraèdriques obtenues par triangulation de Delaunay sur les points échantillons. Ces cellules sont alors sculptées par cohérence avec les silhouettes. L'approche proposée conserve la robustesse d'une approche volumétrique mais s'avère nettement plus avantageuse en précision et en complexité. Elle rend donc possible la modélisation d'objets de géométrie complexe en temps-réel. Les applications à la modélisation d'objets en temps-réel sont discutées et des résultats préliminaires sur des données réelles et de synthèse sont présentés.

Mots Clés

Reconstruction, Images mutiples, Géométrie, Systèmes temps-réel, Modélisation à partir d'images, Enveloppe visuelle

Abstract

This paper addresses the problem of computing visual hulls from image contours. We propose a new hybrid approach which overcomes the precision-complexity trade-off inherent to voxel based approaches by taking advantage of surface based approaches. To this aim, we introduce a space discretization which does not rely on a regular grid, where most cells are ineffective, but rather on an irregular grid where sample points lie on the surface of the visual hull. Such a grid is composed of tetrahedral cells obtained by applying a Delaunay triangulation on the sample points. These cells are carved afterward according to image silhouette information. The proposed approach keeps the robustness of volumetric approaches while drastically improving their precision and reducing their time and space complexities. It thus allows modeling of objects with complex geometry, and it also makes real time feasible for precise models. Applications to real time object modeling are discussed and preliminary results with synthetic and real data are presented.

Keywords

Reconstruction, Multiple Images, Geometry, Real-Time Systems, Modeling from multiple images, Visual Hull

1 Introduction

Plaçons-nous dans un contexte où l'on dispose de plusieurs silhouettes d'un même objet correspondant au points de vue de différentes caméras. L'*enveloppe visuelle* se définit comme la forme maximale cohérente avec les silhouettes de l'objet. Une telle approximation de l'objet rend compte de toute l'information géométrique disponible à partir de ces silhouettes, et présente donc un intérêt pour toute application manipulant celles-ci. Nous décrivons ici une approche pour calculer l'enveloppe visuelle en utilisant efficacement l'information de ces silhouettes. Le but est de proposer une nouvelle méthode à la fois pratique et de complexité spatiale et temporelle raisonnable, fournissant des modèles précis d'objets complexes.

La notion d'enveloppe visuelle a été introduite par Laurentini [13] dans un contexte théorique où un nombre infinis de points de vue entourant l'objet sont considérés. Avant et après cette contribution, l'enveloppe visuelle a été très étudiée, de manière implicite et explicite, dans les communautés de vision et d'image de synthèse. En particulier, il a récemment été montré [14] que l'enveloppe visuelle d'un objet de surface courbe est un polyèdre topologique que l'on peut déterminer avec une calibration faible. Cependant la solution fournie par cet article est peu adaptée à la plupart des situations réelles. Il existe beaucoup d'autres algorithmes fournissant des approximations de l'enveloppe visuelle dans les deux communautés. Certains s'intéressent au volume délimité par l'enveloppe visuelle et se basent sur des discrétisations de l'espace. D'autres visent à reconstruire la surface de l'enveloppe visuelle en fournissant des points isolés ou un maillage.

Les approches volumétriques se basent sur une discrétisation de l'espace en cellules élémentaires, les voxels,

qui sont sculptés au regard de leur projection sur les images et de l'appartenance ou non de celle-ci aux silhouettes de l'objet. Une première approche fut proposée par Martin et Aggarwal [15], qui utilisaient de cellules parallélépipédiques alignées sur les axes. Plus tard une représentation adaptative de l'enveloppe visuelle fut proposée [5] sous forme d'octree. Au cours des années 90, des approches efficaces [21, 18, 4] ont été présentées pour calculer des représentations voxeliques. Les approches mentionnées sont purement géométriques et ne considèrent pas l'information photométrique. Des méthodes récentes [12] l'utilisent en revanche pour sculpter des voxels selon la *cohérence photométrique* de leur projection sur les différentes images. Consulter [19, 8] pour un tour d'horizon des approches volumétriques. Toutes les approches mentionnées se basent sur une grille régulière de voxels et peuvent traiter des objets de géométrie complexe. Cependant ces approches sont coûteuses en ressources et imprécises puisque la plupart des voxels ne rendent pas compte de la surface de l'enveloppe visuelle, qui est l'information utile.

Les approches surfaciques utilisent une stratégie différente. Des éléments de la surface de l'enveloppe visuelle, tels des points ou des facettes, sont estimés par intersection des surfaces des cônes de vue associés au contours occultants. Baumgart [2] utilise pour ce faire une approximation polygonale des contours occultants.[11, 9, 6, 22, 3] s'intéressent à des points isolés reconstruits en utilisant des approximations locales du second ordre de la surface. Plus récemment, des approches reconstruisent des fragments de surface [20], ou des bandes [16] de l'enveloppe visuelle.

Les approches surfaciques peuvent être précises comparées aux approches volumétriques, cependant les modèles produits sont souvent incomplets ou erronés, en particulier si l'on considère des objets complexes. Ces anomalies découlent de la sensibilité aux instabilités numériques de ces algorithmes et des calculs impliquant les cônes de vue, dont le lieu d'intersection est mal défini. Matusik et al. [17] ont montré que l'on peut obtenir de nouvelles images d'un objet grâce à son enveloppe visuelle en n'effectuant que des calculs 2D. Ce résultat intéressant découle de la structure projective de l'enveloppe visuelle précédemment soulignée par [14]. Cependant la méthode proposée ne fournit pas de modèle géométrique explicite comme le requièrent de nombreuses applications.

Notre approche regroupe les avantages des deux catégories de méthodes mentionnées. Elle utilise la robustesse des approches volumétriques tout en conservant la précision inhérente aux approches surfaciques. L'espace est toujours divisé en cellules élémentaires, mais les points échantillonnant l'espace sont pris sur l'enveloppe visuelle plutôt que sur une grille régulière, propre aux méthodes volumétriques antérieures. Les cellules élémentaires sont des tétraèdres obtenus par triangulation de Delaunay appliquée sur ces points. Les cellules sont sculptées par reprojection sur les images. La contribution par rapport aux méthodes antérieures est double: d'une part les points utilisés pour construire le modèle sont sur la surface de l'enveloppe visuelle; d'autre part la surface obtenue découle de la triangulation de Delaunay, un algorithme robuste dont les implémentations sont déjà disponibles et rapides.

L'organisation de l'article est la suivante. Nous introduisons en section 2 les définitions utilisées. La méthode d'obtention des points sur l'enveloppe visuelle est décrite en section 3. L'algorithme utilisé pour obtenir l'approximation surfacique de l'enveloppe visuelle est expliqué en 4. Des résultats expérimentaux sont présentés en section 5. Enfin, les perspectives possibles pour ces travaux seront exposées.

2 Définitions

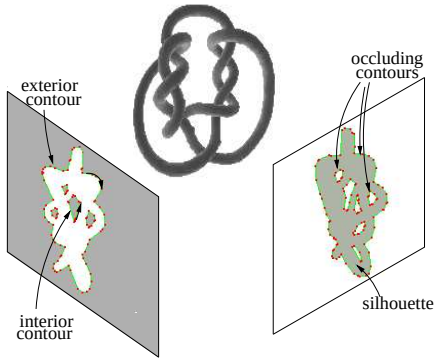


Figure 1: Les contours occultants délimitent la silhouette de l'objet dans le plan image. La région grisée sur la figure de gauche représente les points image qui sont à l'extérieur d'au moins un contour. Son complément, grisé dans l'image de droite, représente les points image qui sont à l'intérieur de tous les contours, et qui appartiennent par conséquent à la silhouette.

Contours Considérons une scène composée de plusieurs objets, observée par un ensemble de caméras à sténopé. Nous supposerons la surface des objets fermée et orientable, courbe ou polyédrique. En outre elle peut être de genre non nul. Les *contours d'occultation* sont le lieu des points sur la surface des objets où les lignes de vue sont tangentes à cette surface. Les contours d'occultation définissent, par projection dans les images, les *contours occultants* [15] qui délimitent la silhouette des objets dans chaque plan image. Nous noterons dans ce qui suit les numéros de contour en indice et les numéros d'image en exposant. Ainsi, $\mathcal{O}_j^i$ désignera le $j$ème contour occultant dans l'image i. Les contours occultants sont orientés dans les images, de telle façon que la silhouette de l'objet qu'elle délimite est localement à gauche du contour. L'orientation

des contours extérieurs et intérieurs de l'objet, respectivement directe et indirecte, découle de cette définition. Nous appellerons *région intérieure* d'un contour occultant la région fermée du plan délimitée par le contour et contenant la silhouette, et nous appellerons *région extérieure* son complément dans le plan image (voir figure 1).

Cônes de vue Intuitivement, le cône de vue associé à un contour occultant est un cône généralisé dont le sommet est le centre optique de la caméra et dont la base est la région intérieure de ce contour. Plus formellement le *cône de vue* $\mathcal{V}_j^i$ associé au contour $\mathcal{O}_j^i$ est la fermeture de l'ensemble des rayons passant par les points de la région intérieure de $\mathcal{O}_j^i$ et par le centre de projection de l'image i. Ainsi $\mathcal{V}_j^i$ est tangent à la surface de l'objet qui lui correspond, le lieu de cette tangence étant le contour d'occultation qui se projette en $\mathcal{O}_j^i$. Selon l'orientation de $\mathcal{O}_j^i$, contour extérieur ou intérieur, le cône de vue $\mathcal{V}_j^i$ est un ensemble respectivement convexe ou non-convexe de $\mathbb{R}^3$. Les surfaces délimitantes des cônes $\mathcal{V}_j^i$ s'intersectent dans l'espace en des courbes dont les points n'appartiennent pas à la surface des objets, à l'exception de certains points appelés *points frontière*, lieux d'intersection des contours d'occultation. Notons que dans le cas de surfaces polyédriques, les points frontière ne sont pas nécessairement isolés et peuvent former des arêtes frontière.

Enveloppe Visuelle L'enveloppe visuelle se définit généralement comme l'intersection de tous les cônes de vue associés aux points de vue considérés, et constitue donc une région fermée de l'espace dont les points se projettent à l'intérieur de tous les contours occultants. Soient $\mathcal{I}$ l'ensemble des images considérées, et $\mathcal{C}$ l'ensemble des contours considérés. L'enveloppe visuelle nous est donnée par la formule suivante:

$$\mathcal{VH}(\mathcal{I},\mathcal{C}) = \bigcap_{i\in\mathcal{I},j\in\mathcal{C}} \mathcal{V}_j^i,$$

où $\mathcal{V}_j^i$ est le cône de vue associé au contour occultant j dans l'image i. Lorsque l'on considère un ensemble fini $\mathcal{I}$ d'images, l'enveloppe visuelle est un polyèdre topologique constitué de fragments de cônes délimités et raccordés par des courbes d'intersection de cônes [14]. En pratique, les contours occultants sont approximés par des courbes polygonales 2D, induisant donc des cônes de vue et une enveloppe visuelle polyédriques. La définition ci-dessus est valable dans le contexte où chaque caméra de $\mathcal{I}$ voit un seul et même objet. Mais elle est incorrecte dans le cas de scènes constituées de plusieurs objets distincts, dont certains n'apparaissent pas dans toutes les images. Nous pourrions combler intuitivement cette lacune en étendant la définition ci-dessus à l'union des enveloppes visuelles de chaque objet pris individuellement. Soient $\mathcal{K}$ l'ensemble des objets de la scène et $\mathcal{C}_k$ l'ensemble des contours de l'objet k, alors:

$$\begin{aligned}\mathcal{VH}(\mathcal{I},\mathcal{K}) &= \bigcup_{k\in\mathcal{K}} \mathcal{VH}(\mathcal{I},\mathcal{C}_k),\\ &= \bigcup_{k\in\mathcal{K}} \Big(\bigcap_{i\in\mathcal{I}_k, j\in\mathcal{C}_k} \mathcal{V}_j^i\Big),\end{aligned} \tag{1}$$

où $\mathcal{I}_k$ est le sous-ensemble des images de $\mathcal{I}$ où l'objet k apparaît. L'utilisation de cette définition requiert la connaissance des ensembles $\mathcal{C}_k$ et $\mathcal{I}_k$, ce qui revient en pratique à identifier tous les contours de l'objet k sur l'ensemble des images. Cette opération est non triviale et est rendue d'autant plus difficile que les silhouettes des différents objets peuvent se chevaucher.
Une autre solution est de définir l'enveloppe visuelle comme étant l'ensemble des points de $\mathbb{R}^3$ se projetant à l'intérieur d'une silhouette dans chaque image où le point est visible. Un premier pas dans cette direction est de considérer l'expression suivante:

$$\mathcal{VH}(\mathcal{I},\mathcal{K}) = \bigcap_{i\in\mathcal{I}} \Big(\bigcup_{k\in\mathcal{S}^i} \Big(\bigcap_{j\in\mathcal{C}_k^i} \mathcal{V}_j^i\Big)\Big), \tag{2}$$

où $\mathcal{S}^i$ est l'ensemble des silhouettes dans l'image i et $\mathcal{C}_k^i$ est l'ensemble des contours associés à la silhouette k dans i. Cette expression revient à appliquer (1) sur un ensemble $\mathcal{K}$ d'objets virtuels dont les silhouettes dans les images sont soit disjointes soit entièrement inclues dans une autre dans toutes les images. L'intérêt réside dans le fait que les contributions de chacun de ces objets est distinguée par ses silhouettes. Comme chaque silhouette est constitué d'exactement un contour extérieur et éventuellement de plusieurs contours intérieurs, l'expression (2) peut facilement s'appliquer en identifiant les contours extérieurs dans l'ensemble des images.
Néanmoins, l'expression 2 n'est pas complètement satisfaisante puisqu'elle ne prend pas en compte le fait que certains objets virtuels peuvent ne pas être vus de toutes les caméras (i.e. $\exists k,i \mid \bigcap_{j\in\mathcal{C}_k^i} \mathcal{V}_j^i = \emptyset$). En conséquence, un objet pour lequel il existe au moins une image où il ne figure pas ne contribue pas à l'enveloppe visuelle(voir figure 2-(b)). Cela est dû au fait que l'intersection des contributions de chaque image dans l'expression 2 devrait être effectuée uniquement dans leur domaine commun de visibilité. Une approche plus simple est de considérer le complémentaire de l'enveloppe visuelle dans ce domaine de visibilité. Il s'agit d'une région ouverte de $\mathbb{R}^3$ définie par:

$$\begin{aligned}\mathcal{VH}^c(\mathcal{I},\mathcal{K}) &= \mathcal{D} \setminus \left[\bigcap_{i\in\mathcal{I}} \Big(\bigcup_{k\in\mathcal{S}^i} \Big(\bigcap_{j\in\mathcal{C}_k^i} \mathcal{V}_j^i\Big)\Big)\right],\\ &= \bigcup_{i\in\mathcal{I}} \Big(\bigcap_{k\in\mathcal{S}^i} \Big(\bigcup_{j\in\mathcal{C}_k^i} \mathcal{D}\setminus\mathcal{V}_j^i\Big)\Big)\end{aligned} \tag{3}$$

où $\mathcal{D}$ est le domaine de visibilité commun à toutes les images dans $\mathbb{R}^3$ et où $\mathcal{D}\setminus\mathcal{E}$ est le complémentaire d'un ensemble donné $\mathcal{E}$ dans ce domaine de visibilité. En utilisant (2), les objets qui n'apparaissent pas dans une image peuvent encore contribuer à l'enveloppe visuelle puisque les contributions vides ne change pas la contribution des autres images dans l'expression ci-dessus. La connaissance de l'enveloppe visuelle ou de son complémentaire est équivalente dans la mesure où la surface qui nous intéresse délimite ces deux régions, et que les opérations d'identification de cellules appartenant à l'enveloppe visuelle ou à son complémentaire sont duales. L'expression

ci-dessus est en fait celle qui est implicitement utilisée par les approches volumétriques pour sculpter les voxels. Notons que l'expression (2) peut également être modifiée pour tenir compte des objets qui ne sont pas toujours visibles (en considérant que le complémentaire du domaine de visibilité de la caméra est une région de contribution de celle-ci), cependant l'utilisation du complémentaire de l'enveloppe visuelle simplifie les expressions et les algorithmes.

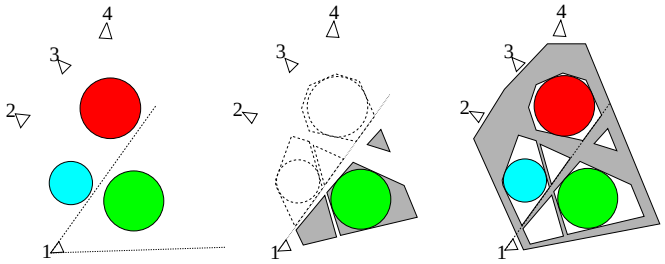


Figure 2: Une situation à 4 points de vue: (a) scène d'origine, la caméra 1 ne voit que l'objet vert; (b) utilisation de l'expression (2): l'enveloppe visuelle (grisée) ne contient aucune contribution relative aux objets rouge et bleu; (c) utilisation de l'expression (3), on calcule le complément de l'enveloppe visuelle, qui inclut les contributions des objets rouge et bleu.

Notons que l'utilisation de (2) et (3) induit l'ajout potentiel d'objets virtuels qui n'apparaissent pas dans la scène d'origine (voir figure 2). Ces objets virtuels indésirables, ou *objets fantômes*, correspondent à des régions de l'espace qui se projettent à l'intérieur d'une silhouette dans toutes les images. La taille et le nombre de ces objets fantômes peuvent généralement être réduits en augmentant le nombre de caméras. De plus, un test de cohérence photométrique pourrait être appliqué sur l'ensemble des images pour les identifier.

3 Points de l'enveloppe visuelle

L'algorithme proposé commence par une étape de détermination de points appartenant à la surface de l'enveloppe visuelle, frontière entre l'enveloppe visuelle et son complément. Ces points sont ensuite utilisés pour partitionner l'espace en cellules dont on identifiera celles qui contribuent à l'enveloppe visuelle. Nous présentons dans cette section la méthode utilisée pour obtenir ces points et discutons de sa complexité.

3.1 Grandes lignes de l'algorithme

Supposons que les contours occultants sont extraits des images. Considérons le contour polygonal $\mathcal{O}_j^i$ dans l'image i. Un point du cône de vue $\mathcal{V}_j^i$ contribue à la surface de l'enveloppe visuelle si: (i) il se projette sur le contour $\mathcal{O}_j^i$, (ii) s'il ne se projette pas à l'intérieur des compléments des silhouettes dans les autres images. Une manière évidente de calculer l'ensemble de ces points consiste donc à prendre des points sur le contour $\mathcal{O}_j^i$ et de considérer l'intersection de leur ligne de vue avec les cônes de vue des autres images. Ces intersections définissent un ou plusieurs intervalles sur la ligne de vue qui correspond à la contribution de cette ligne de vue à la surface de l'enveloppe visuelle. Soit p_j^i un point de $\mathcal{O}_j^i$. Les intervalles de contribution sur la ligne de vue de p_j^i sont délimités par les intersections de cette ligne de vue avec les surfaces des cônes de vue concernés. On peut déterminer ces intervalles en effectuant des calculs d'intersection de droite et de cône dans l'espace. Cependant il est plus judicieux d'effectuer les opérations équivalentes en 2D directement dans les images comme suggéré par [17]. En effet, les points 3D définissant les bornes de l'intervalle sur la ligne de vue de p_j^i sont tels que leurs projections appartiennent à la fois à la ligne épipolaire de p_j^i et au contour occultant concerné (voir figure 3). Ces propriétés sont utilisées dans l'algorithme 1 pour reconstruire des points sur la surface de l'enveloppe visuelle.

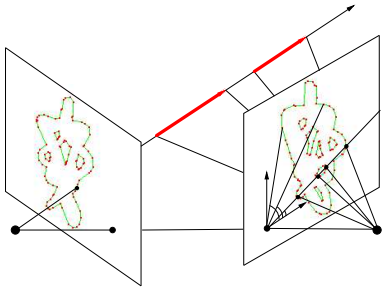


Figure 3: Les intervalles de contribution à l'enveloppe visuelle (en rouge) le long de la ligne de vue. L'angle des lignes épipolaires peut être utilisé pour accélérer la recherche d'intersections entre segments du contour et lignes épipolaires.

Algorithm 1 Visual hull surface points

```
1:  for all contours O_j^i in all images: do
2:    for all images k such that k ≠ i: do
3:      for all points p_j^i in O_j^i : do
4:        compute the epipolar line l of p_j^i in image k,
5:        for all contours O_l^k in image k: do
6:          compute the intersections of l with O_l^k,
7:          update depth intervals along the viewing
            line of p_j^i,
8:        end for
9:      end for
10:     compute the 3D points delimiting intervals along
        the viewing line of p_j^i.
11:   end for
12: end for
```

3.2 Mise à jour des intervalles de profondeur

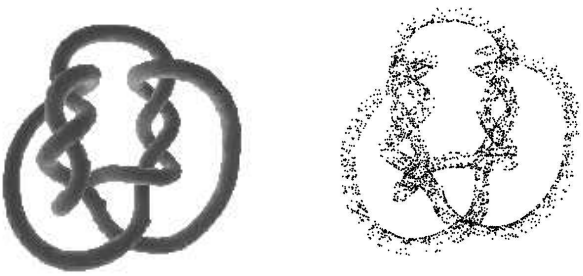
Comme expliqué précédemment, les intersections des droites épipolaires avec les contours occultants sont tout

d'abord calculées. Il est facile d'obtenir à partir de ces intersections la profondeur des points délimitant l'intervalle le long de la ligne de vue, points appartenant à la surface de l'enveloppe visuelle. Il s'agit maintenant de combiner deux listes de profondeurs obtenues avec des contours d'images différentes.

Nous procédons de la manière suivante: l'expression (3) est utilisée pour fusionner sur l'ensemble des contours et des images les intervalles contribuant au complémentaire de l'enveloppe visuelle. L'utilisation de cette formule requiert l'identification de la contribution de chaque silhouette et des différents contours associés. Mais il est possible de travailler directement avec les contours: la contribution des contours intérieurs appartient déjà au complément de l'enveloppe visuelle. C'est pourquoi on peut se contenter d'effectuer l'opération d'intersection sur les contributions des contours extérieurs seulement. Voici l'expression correspondante obtenue à partir de (3):

$$\mathcal{VH}^c(\mathcal{I},\mathcal{K}) = \bigcup_{i\in\mathcal{I}} \left[\left(\bigcap_{j\in \text{Exterieurs}^i} \mathcal{D}\setminus\mathcal{V}_j^i \right) \bigcup \left(\bigcup_{j\in \text{Interieurs}^i} \mathcal{D}\setminus\mathcal{V}_j^i \right) \right] \quad (4)$$

où Exterieurs^i et Interieurs^i sont les ensembles de contours extérieurs et intérieurs de l'image i. L'expression obtenue est équivalente à (3) mais simplifie la fonction de mise à jour des intervalles. Notons que d'après les définitions données de $\mathcal{D}$ et $\mathcal{V}_j^i$, la contribution le long de la ligne de vue du complémentaire de chaque cône de vue doit être limitée à l'intervalle visible depuis l'image qui lui correspond. Ceci permet de rendre compte d'objets non visibles de toutes les caméras, comme précédemment expliqué. La figure (4) montre le résultat de cet algorithme pour un objet de synthèse.



(a) (b)

Figure 4: (a) l'objet *noeuds* extrait du site Web de Hoppe [10];(b) les points de la surface de son enveloppe visuelle pour 40 points de vue localisés sur un cercle autour de l'objet.

3.3 Complexité

Soient n le nombre d'images, m le nombre de contours moyen par image, et q le nombre moyen de points par contour. L'algorithme ci-dessus calcule $\theta(nmq)$ points 3D en un temps $O(n^2m^2qr)$, où r est le majorant de la complexité de la fonction d'intersection ligne-contour[1]. Une implémentation naïve engendrerait un algorithme d'intersection de complexité $l = O(q)$. La complexité asymptotique de l'ensemble serait alors de $O(n^2m^2q^2)$, ou encore $O(N^2)$, où N est le nombre de points 3D calculés.

Il apparaît cependant que la complexité l peut être réduite à $O(1)$. A cet effet il est possible de rectifier l'image k de sorte que les lignes épipolaire deviennent horizontales, auquel cas la recherche d'intersections entre une ligne épipolaire l et le contour occultant $\mathcal{O}_q^k$ est simplifiée en indexant la recherche avec l'ordonnée des épipolaires. Ainsi, déterminer quels segments du contours sont intersectés revient à rechercher ceux dont l'ordonnée des points qui les définissent bornent l'ordonnée de l'épipolaire. On peut utiliser de manière équivalente les angles des droites reliant l'epipole aux points du contour pour indexer la recherche, ce qui évite avantageusement la rectification épipolaire (voir figure 3). Chacune de ces solutions permet d'arriver à $l = O(1)$, moyennant l'ajout d'un nombre d'opérations $\Theta(n^2mq\log mq)$ ou $\Theta(nmq\log mq)$ pour respectivement rectifier les coordonnées de l'image ou calculer les angles pour chaque point du contour, et construire la structure d'indexation dans les deux cas. Notons que [17, 16] utilisent la pente des lignes épipolaires dans un but identique, mais la partition du plan qui en résulte est moins pratique dans la mesure ou une classification erronée peut avoir lieu au voisinage de la droite verticale incidente à l'epipole, endroit où la fonction de calcul de la pente est non monotone et discontinue.

Avec la fonction d'intersection optimisée, la complexité asymptotique de l'algorithme est réduite à $O(n^2m^2q)$. Dans la mesure où l'on a considéré nmq points au total, et compte tenu du fait que chaque point requiert un calcul d'intersection entre sa droite épipolaire et chacun des contours dans chaque image, la borne inférieure de la complexité est de $\Omega(n^2m^2q)$. Ainsi cet algorithme est optimal et calcule $\Theta(nmq)$ points en un temps $O(n^2m^2q)$.

4 Surface de l'enveloppe visuelle

Nous avons précédemment montré comment calculer des points sur la surface de l'enveloppe visuelle. Nous nous intéressons maintenant à l'estimation de la forme de l'enveloppe visuelle. Les approches volumétriques classiques reposaient sur une partition régulière de l'espace en cellules élémentaires: les voxels. Nous proposons ici une partition s'appuyant sur les points calculés ci-dessus, et qui se compose donc de cellules non régulières: les tétraèdres de Delaunay. L'avantage principal réside dans le fait que l'on conserve une forte précision pour une complexité spatio-temporelle raisonnable.

[1] Nous supposons que le nombre d'intersections d'une ligne épipolaire avec un contour occultant est négligeable comparée à n, m, et q, nous considérons ainsi que la fonction de mise à jour des intervalles de profondeur s'exécute en un temps $O(1)$.

4.1 Triangulation des points

L'approche proposée est basée sur la tétraèdrisation de Delaunay des points sur la surface de l'enveloppe visuelle. Les triangulations de Delaunay ont déjà été vastement utilisées pour reconstruire des surfaces 3D à partir de points 3D déstructurés. Ce problème a été exhaustivement étudié dans la précédente décennie, et la plupart des méthodes proposées considèrent que la surface recherchée est contenue dans la triangulation de Delaunay des points. La triangulation de Delaunay comporte deux avantages: d'une part elle assure une partition régulière de l'espace avec des cellules possèdant des propriétés démontrées, telles qu'une boule circonscrite; d'autre part, il en existe des implémentations rapide et robustes.

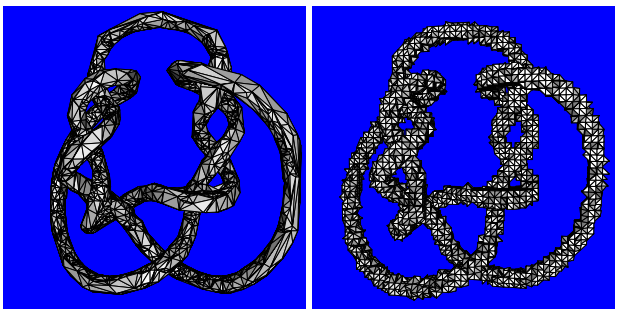
Le problème que nous considérons est semblable mais comporte en entrée, en plus des points 3D, l'information 2D des images. Ainsi notre approche recherche également un sous-ensemble de la triangulation de Delaunay, et le critère permettant de sculpter les cellules tétraèdriques prend cette information supplémentaire en compte. En ce qui concerne la complexité, il est bien connu le cas le pire de la tétraèdrisation de Delaunay est en $O(n^2)$, avec n le nombre de points. En ce qui nous concerne, comme nous calculons $\Theta(nmq)$ points, avec n le nombre d'images, m le nombre moyen de contours, et q le nombre moyen de points par contour, la borne supérieure de la complexité de notre algorithme serait alors de $O(n^2m^2q^2)$, ce qui est supérieur au temps passé à obtenir les points de la surface de l'enveloppe visuelle. Cependant des travaux récents [1] tendent à montrer que la complexité de la triangulation de Delaunay est linéaire pour des points disposés sur un polyèdre. Ceci est par ailleurs confirmé par nos résultats expérimentaux qui montrent que l'algorithme passe la majeure partie de son temps dans la phase de calcul des points. Il est intéressant d'observer que la complexité globale n'est pas dominée par la triangulation de Delaunay.

4.2 Extraction de la surface

La triangulation de Delaunay engendre un ensemble de tétraèdres dont l'union forme l'enveloppe convexe des points fournis. Nous devons maintenant identifier et éliminer ceux qui contribuent au complément de l'enveloppe visuelle. Une méthode ad-hoc consiste à vérifier si la projection du centre des tétraèdres dans les images appartient à une silhouette. Cette approche est rapide dans le cas où l'on dispose d'images binaires indiquant pour chaque pixel son appartenance ou non à l'arrière plan, ce qui est souvent le cas dans le cadre d'applications basées sur les silhouettes. Aussi, notre expérience montre que cette méthode donne des résultats satisfaisants. Notons que des critères d'acceptation plus complexes mettant en jeu la surface ou le volume du tétraèdre sont également envisageables. Nous vérifions actuellement ces possibilités.

5 Résultats expérimentaux

Nous avons appliqué notre méthode à divers jeux de données. Une première expérience consiste à comparer notre approche à l'approche voxelique. La figure 5 montre des résultats obtenus à partir de silhouettes identiques (40 images). Les bornes de la grille des voxels ont été choisies proches de la surface de l'objet, ce qui est rarement le cas pour des applications réelles. Notons que les résultats sont géométriquement meilleurs pour une complexité sensiblement inférieure. Notre modèle comporte en effet 3772 points, alors qu'il faut vérifier 60^3 voxels avec chaque image avec l'autre approche, sans parler d'une éventuelle étape d'extraction de surface. Par ailleurs, le nombre d'images a une influence linéaire sur la borne supérieure de la complexité de l'approche volumétrique, alors que cette influence est quadratique pour notre méthode. En effet le partitionnement de l'espace est indépendant du nombre d'images pour les approches volumétriques, ce qui n'est pas le cas pour notre approche. Notons cependant que l'addition d'images supplémentaires ne permet pas forcément d'améliorer l'estimation de la surface, comme le montrent nos prochaines expériences.



(a) (b)

Figure 5: La surface de l'enveloppe visuelle pour les noeuds: (a) résultat de notre algorithme (3772 points reconstruits) (b) une reconstruction voxelique sur une grille de 60x60x60.

Notre seconde série de résultats a été obtenue à partir d'images d'un tore de synthèse et de points de vue aléatoirement distribués sur une sphère entourant ce tore. La figure 6 montre différentes enveloppes visuelles du tore obtenues en faisant varier le nombre de points échantillonnés sur les contours 2D des images ainsi que le nombre de caméras. Notons que le temps d'exécution de l'algorithme est en $O(n^2m^2q)$ avec n, m, q le nombre d'images, de contours par images, et de points par contour respectivement. L'addition de points échantillonnés sur les contours 2D a donc un impact moins sensible sur le temps d'exécution. Il est surprenant de constater qu'à partir d'un certain nombre d'images ajoutées, la précision du modèle d'enveloppe visuelle calculé décroît. Cela apparaît clairement dans la colonne de gauche, au passage de 16 à 32 images. Ce comportement s'explique en notant que l'ajout d'images multiplie le nombre de points obtenus sur l'enveloppe vi-

suelle. Les tétraèdres obtenus par Delaunay sont donc de plus en plus petits et le calcul de la reprojection de leur centre devient de plus en plus sensible aux diverses erreurs numériques, et en particulier aux erreurs de discrétisation. La parade à ce défaut est donc d'imposer l'augmentation de points échantillonnés sur le contour 2D lorsque le nombre d'images augmente, pour réduire cette erreur. Il est intéressant de constater que cela suggère l'existence d'un rapport optimal entre le nombre d'images et le nombre de points pris sur les contours.

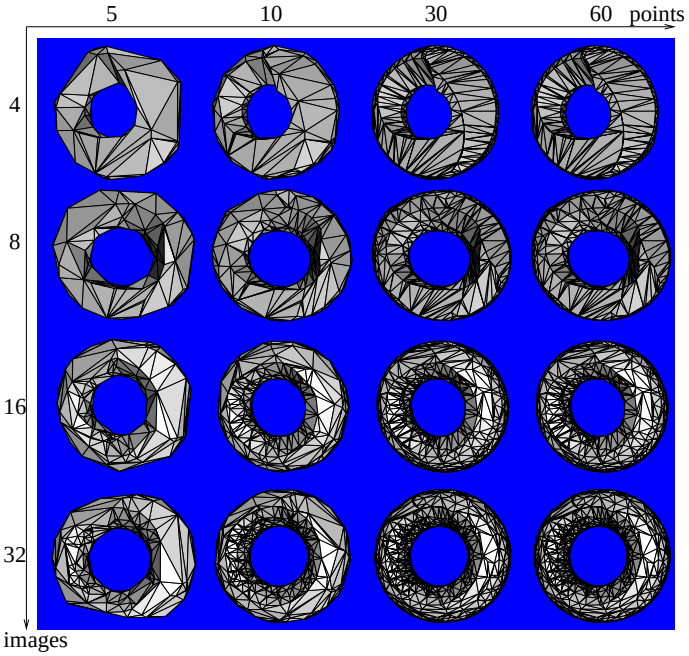


Figure 6: Enveloppes visuelles d'un tore: en abscisse le nombre de points pris sur le contour et en ordonnée le nombre d'images. Les points sont régulièrement échantillonnés sur les contours et les images aléatoirement choisies sur une sphère entourant le tore.

En ce qui concerne le temps d'exécution, facteur crucial pour les applications temps-réel, notre approche calcule le premier modèle de tore en 0.008s et le dernier modèle en 1.3s sur un seul PC à 1.8GHz. Cependant les résultats sur un seul PC avec plus de 4 caméras temps-réel ne sont pas vraiment significatifs en pratique. Sur le modèle des noeuds, et avec 4 caméras aléatoirement placées, le temps d'exécution moyen est de 0.45s pour un modèle final comportant 2200 points. Comme il a précédemment été expliqué, l'algorithme passe la majorité de son temps dans la phase de calcul des points sur la surface de l'enveloppe visuelle, et pas sur la triangulation de Delaunay, qui traite les 2200 points du modèle en 0.3s. En outre la complexité linéaire de cette opération est vérifiée dans nos résultats.

La figure 7 montre des résultats obtenus avec des objets réels. Les contours polygonaux sont extraits dans les images en utilisant l'algorithme optimal de reconnaissance de segments décrit dans [7]. L'exemple d'une personne est intéressant dans la mesure où il fait apparaître les objets fantômes décrits dans la section 2. L'autre exemple dans la figure 8 illustre le potentiel de cette méthode dans le cadre de l'estimation de surfaces courbes. Nous expérimentons dans un cadre comparable en utilisant une grappe de PCs dans un contexte temps-réel, nos résultats préliminaires montrent que l'on peut espérer obtenir des modèles relativement précis en temps-réel.

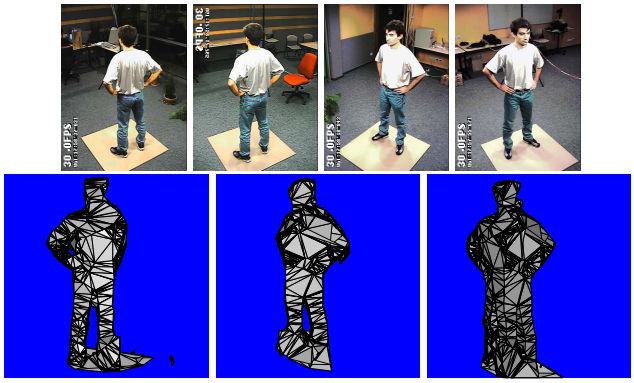


Figure 7: L'enveloppe visuelle d'une personne avec 4 points de vue. Elle contient toute l'information géométrique disponible à partir des silhouettes. Notons toutefois l'apparition de jambes fantômes dans l'image de droite. Ceci est dû aux positions particulières des caméras, et illustre nos remarques en section 2.

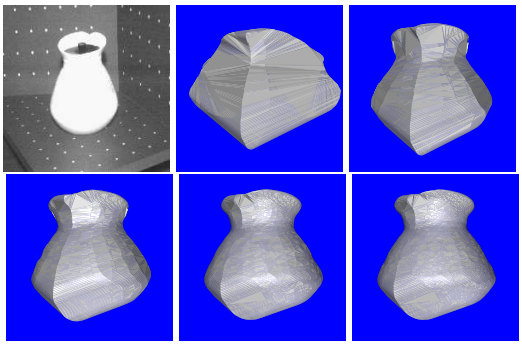


Figure 8: L'enveloppe visuelle d'une cruche avec 2, 5, 10, 20 et 30 images. Certains cônes grossiers sont encore visibles même avec 30 images du fait du manque de vues dans une large région derrière l'objet.

6 Conclusion

Dans cet article, nous avons présenté une approche permettant d'obtenir l'enveloppe visuelle de scènes complexes en disposant des silhouettes de celle-ci. Notre principale contribution est de proposer un algorithme hybride qui tire avantage des approches volumétriques et des approches surfaciques. L'algorithme commence par calculer des points sur la surface de l'enveloppe visuelle, puis extrait la surface de l'enveloppe visuelle en partant d'une triangulation de Delaunay.

La première étape est réalisée en calculant des points délimitant le complément de l'enveloppe visuelle. La deuxième étape consiste à prendre la surface délimitant l'ensemble des tétraèdres dont la projection se trouve à l'intérieur des silhouettes. Nous avons montré que notre approche est équivalente aux approches volumétriques en terme d'efficacité. Chacune de ces deux approches se base sur la même définition de l'enveloppe visuelle et utilise toute l'information géométrique disponible à partir des silhouettes. Cependant nous avons aussi montré que notre approche donne des résultats significativement meilleurs en terme de précision, tout en ayant une complexité plus réduite dans le temps et l'espace. La méthode de reconstruction en résultant se destine donc naturellement aux applications de modélisation en temps-réel.

Nous travaillons actuellement sur des améliorations possibles, et recherchons également d'autres terrains d'application pour notre méthode. D'une part, la classification des tétraèdres se fait au regard de la position du centre projeté de ceux-ci dans les images. Ceci peut être amélioré par l'utilisation d'autres logiques d'élimination. D'autre part, la triangulation de Delaunay est appliquée uniquement sur des points, alors que nous disposons d'une information supplémentaire sur l'enveloppe visuelle, puisque l'ensemble des points de chaque intervalle calculé en section 3 appartient à la surface de l'enveloppe visuelle et en constitue une arête, dont la prise en compte contribuerait à améliorer le modèle final. Enfin, l'implémentation d'applications temps-réel de tels algorithmes de modélisation reste un problème difficile, en particulier dans le contexte de l'utilisation d'une grappe de PCs.

References

[1] Dominique Attali et Jean-Daniel Boissonnat. A linear bound on the complexity of the delaunay triangulation of points on polyhedral surfaces. Rapport de recherche 4453, INRIA, 2002.

[2] B.G. Baumgart. A polyhedron representation for computer vision. In *AFIPS National Computer Conference*, 1975.

[3] E. Boyer et M.-O. Berger. 3D surface reconstruction using occluding contours. *International Journal of Computer Vision*, 22(3):219–233, 1997.

[4] K.M. Cheung, T. Kanade, J.Y. Bouguet, et M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, pages 714 – 720, June 2000.

[5] C.H. Chien et J.K. Aggarwal. Volume/surface octress for the representation of three-dimensional objects. *Computer Vision, Graphics and Image Processing*, 36(1):100–113, 1986.

[6] R. Cipolla et A. Blake. Surface Shape from the Deformation of Apparent Contours. *International Journal of Computer Vision*, 9:83–112, 1992.

[7] I. Debled-Rennesson et J.P. Reveillès. A linear algorithm for segmentation of digital curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(4):635–662, 1995.

[8] C.R. Dyer. Volumetric Scene Reconstruction from Multiple Views. In L.S. Davis, editor, *Foundations of Image Understanding*, pages 469–489. Kluwer, Boston, 2001.

[9] P.J. Giblin et R. Weiss. Reconstruction of Surfaces from Profi les. In *Proceedings of the First International Conference on Computer Vision, London*, pages 136–144, 1987.

[10] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, et W. Stuetzle. Surface reconstruction from unorganized points. In *ACM Computer Graphics (Proceedings SIGGRAPH)*, volume 26(2), pages 71–78, July 1992.

[11] J.J. Koenderink. What Does the Occluding Contour Tell us About Solid Shape? *Perception*, 13:321–330, 1984.

[12] K. Kutulakos et S. Seitz. A Theory of Shape by Space Carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[13] A. Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on PAMI*, 16(2):150–162, February 1994.

[14] S. Lazebnik, E. Boyer, et J. Ponce. On How to Compute Exact Visual Hulls of Object Bounded by Smooth Surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Kauai, (USA)*, volume I, pages 156–161, December 2001.

[15] W.N. Martin et J.K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Transactions on PAMI*, 5(2):150–158, 1983.

[16] W. Matusik, C. Buehler, et L. McMillan. Polyhedral Visual Hulls for Real-Time Rendering. In *Eurographics Workshop on Rendering*, 2001.

[17] W. Matusik, C. Buehler, R. Raskar, S. Gortler, et L. McMillan. Image Based Visual Hulls. In *ACM Computer Graphics (Proceedings Siggraph)*, pages 369–374, 2000.

[18] W. Niem. Automatic Modelling of 3D Natural Objects from Multiple Views. In *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg, Germany*, 1994.

[19] G. Slabaugh, B. Culbertson, T. Malzbender, et R. Schafe. A Survey of Methods for Volumetric Scene Reconstruction from Photographs. In *International Workshop on Volume Graphics*, 2001.

[20] S. Sullivan et J. Ponce. Automatic Model Construction, Pose Estimation, and Object Recognition from Photographs using Triangular Splines. *IEEE Transactions on PAMI*, pages 1091–1096, 1998.

[21] R. Szeliski. Rapid Octree Construction from Image Sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, 1993.

[22] R. Vaillant et O. Faugeras. Using Extremal Boundaries for 3-D Object Modeling. *IEEE Transactions on PAMI*, 14(2):157–173, February 1992.