

Enveloppes Visuelles et Reconstruction 3D

Louise Benoît

Octobre 2009

Introduction au domaine de recherche

Sous la direction de Jean Ponce et Francis Bach.

Table des matières

1	Introduction	3
2	Enveloppes Visuelles	4
2.1	Définition	4
2.2	Enveloppes visuelles polyédrales	6
2.3	Enveloppes visuelles calculées dans l'espace des images	7
2.4	Enveloppes visuelles projectives	8
3	Taraudage	8
3.1	Cadre d'étude	8
3.2	Algorithme	9
3.3	Résultats	10
4	Fusion des deux approches	10
4.1	Idées et difficultés	11
4.2	Résultats	12
5	Conclusion	13

1 Introduction

La vision par ordinateur

La vision par ordinateur est la science de la vision des machines. C'est une discipline scientifique qui s'intéresse à la construction de systèmes artificiels qui permettent d'obtenir des informations à partir d'images. Les données d'entrée peuvent prendre de nombreuses formes : photographies, séquences vidéos, images de caméras multiples ou données multi-dimensionnelles d'un scanner médical, etc. Des sous-domaines de la vision par ordinateur sont par exemple la reconstruction de scènes, la détection d'évènements, la reconnaissance d'objets, l'apprentissage et la restauration d'images.

Même si des travaux plus anciens existent, ce n'est pas avant la fin des années 70 qu'une étude plus approfondie a commencé. Souvent, les questions soulevées ont été originaires d'un autre domaine. C'est pourquoi il n'y a en général pas de méthode standard pour résoudre les problèmes, et qu'au lieu de cela, il existe une abondance de méthodes pour résoudre des tâches bien définies.

Il y a bien entendu de multiples domaines d'applications à la vision par ordinateur. Parmi les plus importants, on peut citer le traitement d'images médicales, de nombreuses applications militaires, les effets spéciaux au cinéma, la surveillance, etc.

Décrivons rapidement les tâches principales de la vision par ordinateur.

- Reconnaissance d'objets

Un problème classique est de déterminer si un objet particulier se trouve ou non dans une image. On peut distinguer la détection (trouver l'emplacement d'un visage dans une image) et la reconnaissance (trouver le visage d'une personne en particulier). Ces deux tâches font appel à des outils très différents.

- Restauration d'images

Le but de la restauration d'images est d'éliminer le bruit des images. L'approche la plus simple pour cela est de créer un filtre. Des modèles plus sophistiqués utiliseront la structure de l'image pour la débruiter.

- Reconstruction de scènes

Étant données une ou plusieurs images d'une scène, on cherche à reconstruire un modèle *3D* de la scène. C'est plus précisément ce problème auquel je m'intéresse.

La reconstruction 3D

Malgré la richesse d'informations contenues dans une photographie, la profondeur d'un point de la scène n'est pas directement accessible dans une unique image. Par contre, dès que l'on dispose de plusieurs images prises sous des angles de vues légèrement différents, il devient possible de calculer la position spatiale des points observés dans au moins deux images. On désigne alors par *stéréovision* le processus qui permet de combiner entre elles plusieurs images d'une même scène pour en extraire des informations géométriques tridimensionnelles.

Le processus stéréoscopique le plus simple n'utilise que deux images : il s'agit alors de stéréovision binoculaire. En effet, la perception humaine du relief se forme dans le cerveau lorsqu'il reconstitue une seule image à partir de la perception des deux images planes et différentes provenant de chaque œil. Mais en pratique, on utilise bien souvent plus de deux caméras pour faire une reconstruction tridimensionnelle.

2 Enveloppes Visuelles

Plaçons-nous dans le cadre général suivant : on considère un ensemble de photographies d'une même scène prises depuis des points de vues connus mais arbitrairement distribués. Le but final est de reconstruire un modèle $3D$ de la scène initiale à partir des photographies. Comme il est plus simple de raisonner en dimension 2, nous nous placerons parfois dans le cas où les objets sont contenus dans un plan, qui contient également les caméras qui observent la scène. Dans ce cas, les "images" sont de dimension 1, et on souhaite reconstruire une scène de dimension 2.

Dans cette partie, nous allons commencer par définir la notion d'enveloppe visuelle, première étape de cette reconstruction, qui est déterminée à partir des silhouettes de la scène. Nous verrons ensuite plusieurs algorithmes de construction de ces enveloppes visuelles.

2.1 Définition

Les silhouettes ont été utilisées pour la première fois par Baumgart dans l'article [1], où il propose de calculer des approximations de formes polyédrales en intersectant les cônes engendrés par les silhouettes. Le terme "enveloppe visuelle" fut introduit plus tard par Laurentini (voir [2]) pour décrire le volume maximal compatible avec un ensemble de silhouettes. Le but de cette partie est d'utiliser l'information contenue dans les silhouettes pour distinguer des régions de l'espace où un objet est ou n'est pas présent. L'ultime résultat est une forme appelée enveloppe visuelle de l'objet. L'enveloppe visuelle contient toujours l'objet.

Supposons maintenant qu'un objet ($2D$ ou $3D$) est vu par un ensemble R de vues de référence. L'objet se projette dans chaque vue r en une silhouette s_r . Pour chaque vue r , on considère le cône v_r défini par les rayons issus du centre de la caméra correspondante, et passant par les pixels de la silhouette s_r .

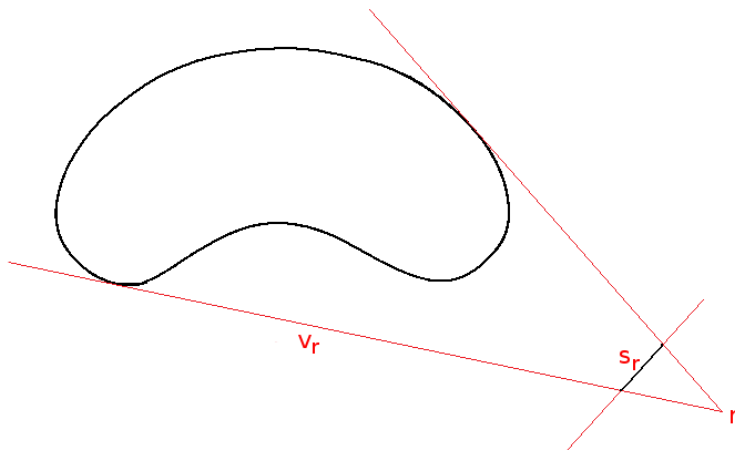


FIG. 1 – Un objet $2D$ représenté avec une vue, sa silhouette et le cône correspondant.

L'objet est toujours contenu dans le cône v_r . Ceci est vrai pour chaque vue r , donc l'objet est contenu dans $v_R = \bigcap_{r \in R} v_r$. Quand la taille de R tend vers l'infini et inclut

par conséquence toutes les vues possibles, v_R converge vers une forme appelée l'*enveloppe visuelle* de l'objet. L'enveloppe visuelle n'est pas égale à l'objet lui-même car certaines zones concaves ne peuvent pas être prises en compte quand on utilise seulement les silhouettes.

Plus concrètement, si la scène est de dimension 2, l'enveloppe visuelle est l'enveloppe convexe de l'objet, et si la scène est de dimension 3, l'enveloppe visuelle est contenue dans l'enveloppe convexe.

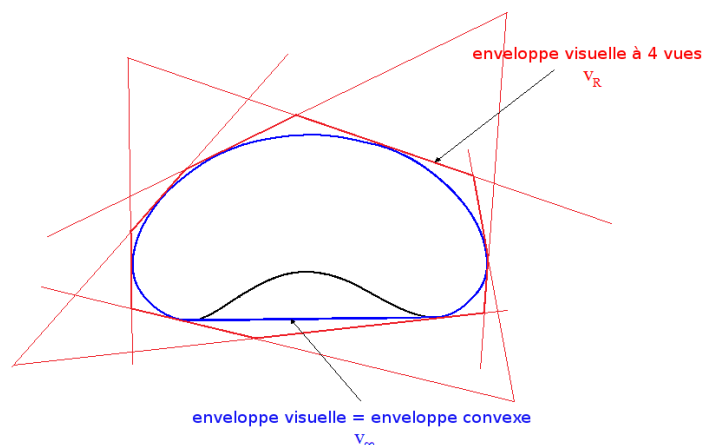


FIG. 2 – v_∞ et v_R pour un objet 2D : v_∞ est l'enveloppe convexe de l'objet.

Dans la pratique, on ne peut construire que des approximations de l'enveloppe visuelle, en utilisant un nombre fini de vues. Étant donné un ensemble de vues R , l'approximation v_R est la meilleure description géométrique qui puisse être obtenue simplement avec les silhouettes de la scène. La figure 2 montre v_∞ et v_R dans un cas où R contient 4 vues différentes d'un unique objet 2D.

La figure 3 montre l'une des silhouettes d'un objet tridimensionnel. Cette silhouette est la projection de l'enveloppe visuelle (autant v_∞ que v_R). On peut voir que la silhouette est contenue dans la projection de l'enveloppe convexe de l'objet.

Les enveloppes visuelles peuvent être calculées à partir des silhouettes des objets dans des images, et un certain nombre d'approches ont été proposées. Nous pouvons globalement les séparer en deux catégories : les volumétriques et les surfaciques. La première catégorie inclut des méthodes qui approximent l'enveloppe visuelle par un ensemble de voxels (voir [3] et [4]). Ces approches reposent sur une grille régulière de voxels et sont robustes à des topologies complexes. Néanmoins, la discrétisation de l'espace utilisée entraîne des approximations, et la précision des résultats est décevante comparée à la complexité des algorithmes. Les approches surfaciques estiment au contraire l'enveloppe visuelle en intersectant les cônes issus des silhouettes. Je vais maintenant décrire trois types d'approches surfaciques.

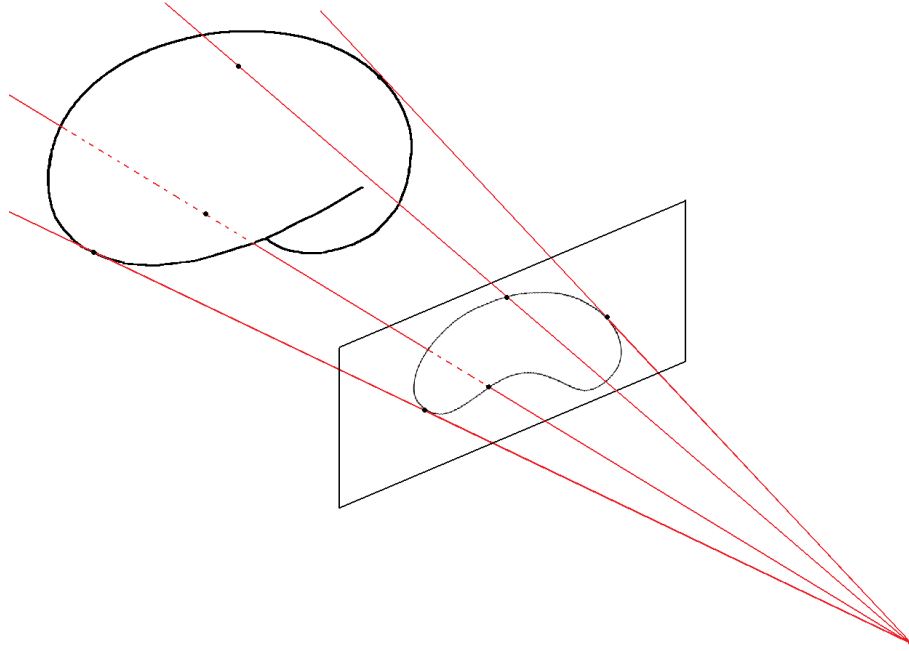


FIG. 3 – Silhouette d’un objet $3D$: elle est incluse dans la projection de l’enveloppe convexe.

2.2 Enveloppes visuelles polyédrales

Expliquons tout d’abord un premier type d’approche surfacique où l’on suppose que chaque silhouette est constituée d’un ensemble de polygones $2D$. Ainsi, l’enveloppe visuelle que l’on cherche à construire sera un polyèdre $3D$.

Il existe des travaux ([5, 6]) sur les opérations booléennes de polyèdres $3D$. La plupart de ces algorithmes nécessitent de décomposer les polyèdres de départ en polyèdres convexes. Ensuite les opérations sont effectuées sur ces polyèdres convexes. Au contraire, l’algorithme décrit dans l’article [7] ne fait aucune hypothèse de convexité ; il exploite plutôt le fait que les objets intersectés sont des cônes, i.e. que leur section est constante à un facteur d’échelle près. Décrivons plus précisément les idées de l’article [7].

Dans cet algorithme, nous supposons donc que chaque silhouette est un ensemble de polygones $2D$, non nécessairement convexes. Ces polygones peuvent avoir des trous. Chaque polygone consiste en un ensemble d’arêtes qui définissent son contour. Pour calculer l’enveloppe visuelle en utilisant les silhouettes, il faut calculer les intersections des cônes définies par ces silhouettes. Le polyèdre qui en résulte est décrit par l’ensemble de ses faces. Les faces de ce polyèdre sont forcément contenues dans les faces des cônes issus des arêtes des silhouettes.

L’algorithme peut être décrit ainsi : pour chaque silhouette et pour chaque arête de ces silhouettes, on calcule la face du cône correspondante. Ensuite, on intersecte cette face avec les cônes des autres silhouettes. Le résultat de cette intersection est un ensemble de polygones qui décrivent la surface de l’enveloppe visuelle à n vues.

L’intersection d’une face d’un cône avec les autres cônes est une opération dans l’espace tridimensionnel, mais ces intersections peuvent être ramenées à des intersections plus

simples en dimension 2, comme nous le verrons dans la partie suivante. L'article [7] montre comment calculer efficacement l'intersection des cônes projetés avec les silhouettes : le processus est accéléré en transformant préalablement les silhouettes en structures "Edge-Bin". La structure "Edge-Bin" partitionne la silhouette dans l'espace de façon à calculer rapidement l'ensemble des arêtes qui intersecte la projection d'une face d'un cône.

Enfin, suite aux intersections successives de paires de cônes, on obtient $n - 1$ ensembles de polygones pour chaque face de chaque cône, où n est le nombre de silhouettes. Les faces de l'enveloppe visuelle sont les intersections de ces ensembles de polygones pour chaque face de cône. L'article [7] utilise un algorithme qui produit directement des triangles.

L'un des intérêts de cette méthode est qu'elle produit un modèle tridimensionnel explicite de l'enveloppe visuelle. Néanmoins, l'algorithme calcule des bandes disjointes qui sont des sous-ensembles de l'enveloppe visuelle. Un défaut de cette méthode est de multiplier les intersections de cônes et de nécessiter une opération supplémentaire pour raccorder les différents morceaux, sans aucune garantie topologique. C'est pour cela que les articles [8] et [9] montrent comment calculer l'enveloppe visuelle polyédrale dans son ensemble avec un nombre réduit d'opérations. L'article [8] distingue deux étapes : une première étape calcule les arêtes visibles de l'enveloppe visuelle, et la deuxième estime les faces par une triangulation de Delaunay. L'article [9] améliore cette approche en construisant, avec un algorithme moins coûteux (en temps), le polyèdre exact qui est cohérent avec les silhouettes. Pour ce faire, il remplace la seconde étape mentionnée ci-dessus par un algorithme qui reconstruit directement un maillage connexe.

2.3 Enveloppes visuelles calculées dans l'espace des images

Détaillons maintenant une méthode efficace décrite dans l'article [11] qui permet le calcul d'une enveloppe visuelle pour n vues à partir des silhouettes correspondantes. L'algorithme décrit dans cet article calcule l'enveloppe visuelle exacte sans construire de représentation auxiliaire géométrique ou volumétrique. L'article présente la particularité de raisonner dans l'espace image pour chacune des étapes de construction. La représentation qui en résulte dépend du point de vue.

Étant donnée une vue, le but est de calculer l'intersection de chaque rayon avec l'enveloppe visuelle. Le calcul d'une enveloppe visuelle ne demande que de faire des intersections, et comme les cônes ont une coupe fixée (à échelle près), les intersections tridimensionnelles des rayons peuvent être ramenées à des intersections moins coûteuses en $2D$. Le calcul nécessite trois étapes :

1. On projette le rayon $3D$ dans une image de référence.
2. On calcule l'intersection du rayon projeté avec la silhouette $2D$. Cette intersection fournit un ensemble d'intervalles le long du rayon qui sont à l'intérieur du cône.
3. Chaque intervalle est ensuite relevé dans la dimension supérieure grâce à une fonction projective, puis est intersecté avec les résultats des autres images de référence.

La figure 4 qui provient de l'article [11] illustre ce processus.

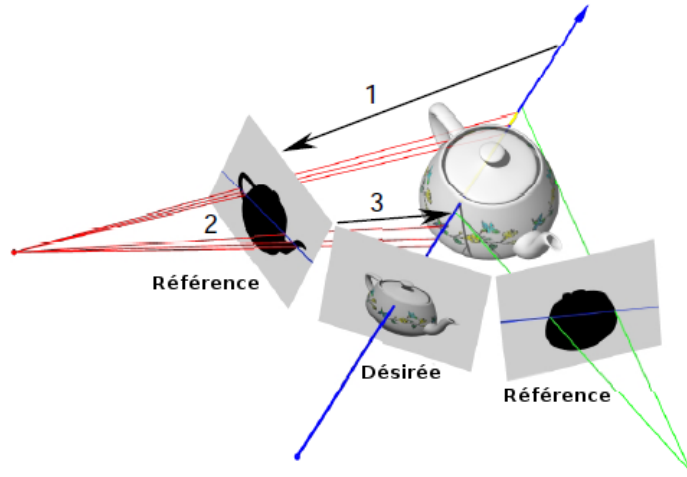


FIG. 4 – Les trois étapes de construction de l’enveloppe visuelle selon l’article [11].

2.4 Enveloppes visuelles projectives

Enfin, l’article [12] se place dans le cas où les objets observés sont lisses, et utilise le cadre mathématique de la géométrie sphérique pour construire l’enveloppe visuelle de la scène. L’algorithme détaillé dans cet article comporte deux étapes.

Dans un premier temps, il s’agit de construire le 1–squelette de l’enveloppe visuelle. Ce squelette est formé de sommets, qui sont les intersections de trois cônes et les points de tangence double, et les arêtes, intersections de deux cônes, sont des courbes lisses qui relient deux sommets. Pour construire ce squelette, l’algorithme ajoute les caméras les unes après les autres, et met à jour les informations fournies par les caméras au fur et à mesure.

Ensuite, l’algorithme reconstruit les faces de l’enveloppe visuelle en triangulant les bandes délimitées par les arêtes du squelette. Pour cela, on établit un ordre sur les paires de points situées sur deux arêtes.

L’un des intérêts de cette méthode est de pouvoir améliorer au fur et à mesure la précision de l’enveloppe visuelle en ajoutant davantage de caméras. De plus, la topologie du résultat est maîtrisée.

3 Taraudage

Le but de l’algorithme de Taraudage (ou "Space Carving" dans le texte, c’est à dire littéralement d’évidage de l’espace), est de reconstruire une scène $3D$ à partir d’une famille de photographies de la scène. Pour cela, on va partir d’une forme contenant la scène, et on va supprimer itérativement les voxels jusqu’à converger vers la reconstruction voulue.

3.1 Cadre d’étude

Kutulakos et Seitz considèrent dans l’article [13] le problème de reconstruction d’une forme tridimensionnelle à partir d’un ensemble de photographies prises depuis des points

de vue arbitrairement distribués mais connus. On suppose que la silhouette dans chaque image est connue.

L'article [13] utilise un modèle d'illumination pour décrire la radiance de la scène : les auteurs attribuent une fonction radiance à l'objet. À chaque point de la surface et à chaque direction de l'espace est associée la couleur de la lumière reflétée dans cette direction pour le point considéré. Cette fonction décrit ainsi l'illumination de la surface. Cela permet alors de définir la photo-cohérence : un point $3D$ est photo-cohérent avec une photographie donnée s'il ne se projette pas sur un pixel d'arrière plan et si la couleur de la projection est égale à la valeur de la radiance du point pour la direction correspondante. Une forme et sa radiance associée sont photo-cohérentes avec une photographie si tous les points visibles par la caméra sont photo-cohérents. Enfin, une forme est photo-cohérente avec un ensemble de photographies s'il existe une affectation de fonction radiance aux points visibles qui rend la forme photo-cohérente pour toutes les photographies.

L'article [13] se propose donc de répondre aux trois questions suivantes :

- Comment caractériser l'ensemble de toutes les formes photo-cohérentes ?
- Comment calculer l'une des formes de cet ensemble ?
- Quel est le lien entre la forme choisie et les autres formes photo-cohérentes ?

En effet, le problème de la reconstruction $3D$ est généralement "mal posé" dans le sens où il existe plus d'une forme cohérente avec l'ensemble des images. Cet algorithme traite cette ambiguïté en calculant l'unique reconstruction photo-cohérente de la scène qui contient toutes les autres reconstruction photo-cohérentes. La forme $3D$ obtenue peut servir de représentant à la classe d'équivalence des formes photo-cohérentes.

Comme nous avons pu le voir, l'enveloppe visuelle utilise uniquement l'information correspondant aux pixels d'arrière plan. Au contraire, définissons maintenant la notion d'enveloppe photo, qui utilise cette fois les contraintes supplémentaires dues à la couleur de la lumière reflétée. Le "théorème de l'enveloppe photo" de l'article [13] affirme que la réunion de toutes les formes photo-cohérentes est photo-cohérente. Nous donnons le nom d'enveloppe photo de la scène cette réunion.

L'enveloppe photo représente la reconstruction la plus fine (que l'on puisse faire) sans faire d'hypothèse supplémentaire. Cette enveloppe n'est pas nécessairement fermée, ainsi ce n'est pas une forme au sens strict, mais l'article [13] suppose que c'est toujours le cas.

3.2 Algorithme

Décrivons maintenant l'algorithme de Taraudage. Étant donné un volume initial quelconque contenant la scène, l'algorithme procède en retirant successivement des morceaux de ce volume jusqu'à ce qu'il converge vers l'enveloppe photo. Le choix du volume initial a un impact considérable sur le résultat du processus de reconstruction. Une possibilité est d'initialiser l'algorithme avec l'enveloppe visuelle décrite ci-dessus. L'article [13] choisit comme volume initial un volume fini arbitraire qui contient la scène. Ce volume est représenté par un ensemble fini de voxels. Avec cette représentation, chaque itération d'évidage enlève un unique voxel.

Le "théorème du sous-ensemble" de l'article [13] affirme que si un point de la surface n'est pas photo-cohérent, alors aucun sous-ensemble photo-cohérent de cette forme ne contient ce point. D'après ce résultat, un voxel non photo-cohérent est enlevé à chaque itération. Ainsi, si seuls des pixels non photo-cohérents sont enlevés, le volume sculpté

converge vers l'enveloppe photo. L'ordre dans lequel les voxels sont examinés et enlevés n'est pas important pour garantir la convergence. La forme voulue est atteinte quand on ne peut plus trouver de voxels non-cohérent sur la surface du volume sculpté.

L'article [13] présente ensuite deux implémentations de l'algorithme de Taraudage, qui permettent de calculer efficacement l'enveloppe visuelle avec un minimum d'espace mémoire utilisé.

3.3 Résultats

Voici quelques résultats tirés de l'article [13] : la figure 5 montre 6 images parmi les 16 d'une sculpture de gargouille. La figure 6 montre deux nouvelles vues de la reconstruction tridimensionnelle calculées avec l'algorithme de Taraudage.



FIG. 5 – 6 des 16 images de gargouilles.

4 Fusion des deux approches

Il existe de nombreuses façons de déterminer une forme tridimensionnelle à partir d'images bidimensionnelles. L'article [14] propose une approche probabiliste qui s'intéresse à la reconstruction de cette forme à partir d'une seule image. Pour chaque classe d'objets, la méthode construit un modèle qui fournit une information sur la distribution tri-dimensionnelle moyenne des objets. L'algorithme sélectionne ensuite une classe d'objets et une pose qui maximisent la log-vraisemblance du modèle graphique.

Dans cette partie, je vais m'intéresser plus en détail à une méthode qui reconstruit un modèle tridimensionnel à partir de plusieurs images, en combinant les méthodes de construction des enveloppes visuelles et les idées de l'algorithme de Taraudage.



FIG. 6 – Deux nouvelles vues de la reconstruction tridimensionnelle calculées avec l'algorithme de Taraudage (article [13]).

4.1 Idées et difficultés

Dans l'article [15], les auteurs décrivent une approche qui nécessite un traitement préalable sur les images. En effet, l'utilisateur doit commencer par identifier des régions polygonales dans chaque image. Il attribue ensuite à chaque région une étiquette de telle sorte que deux régions avec la même étiquette dans deux images différentes correspondent au même objet dans la scène 3D photographiée. À partir de ces données, on cherche à reconstituer le plus précisément possible l'enveloppe photo de la scène.

Le partage des photographies en régions polygonales induit une partition de l'espace en cellules. C'est sur ces cellules et non sur des voxels que l'on va utiliser la méthode de l'algorithme de Taraudage. Comme dans l'article [13], l'algorithme est initialisé avec n'importe quel ensemble de cellules dont on sait qu'il contient la scène. L'idée que l'on exploite ensuite est que si l'ensemble contient une cellule non cohérente avec les images 2D, alors aucun sous-ensemble contenant cette cellule ne sera cohérent, et donc la cellule peut être enlevée. L'algorithme de l'article [15] examine donc les cellules les unes après les autres, et enlève au fur et à mesure les cellules incohérentes. Le "Théorème de reconstruction cellulaire" (article [15]) nous assure que le processus converge vers l'enveloppe photo de la scène.

Un point qui reste à préciser dans cette approche est la justification rigoureuse du fait qu'une cellule est "vide". En effet, il faudrait commencer par définir proprement les notions de cellule "vide" ainsi que la notion de "visibilité" : pour obtenir des informations sur une cellule donnée, il faut commencer par déterminer les caméras qui "voient" cette cellule. On pourra ensuite décider si elle est vide ou non.

De plus, la géométrie change à chaque itération, donc des nouvelles cellules peuvent devenir visibles à chaque pas de l'algorithme. Il s'ensuit que l'ordre dans lequel on examine les cellules est important.

4.2 Résultats

L'un des avantages de cette méthode est que la reconstruction obtenue est une réunion de cellules et on peut donc la trianguler afin de recoller la texture.

Voici quelques résultats fournis par l'article [15]. La figure 7 montre quelques images d'une scène contenant un bol et un bagel. C'est à partir de ces images que l'on applique l'algorithme. La figure 8 montre à son tour quelques nouvelles vues de la scène d'une part pour l'enveloppe visuelle, et d'autre part pour la reconstruction finale. On voit notamment que le bol a été creusé par l'algorithme ce qui n'est pas le cas avec seulement l'enveloppe visuelle.

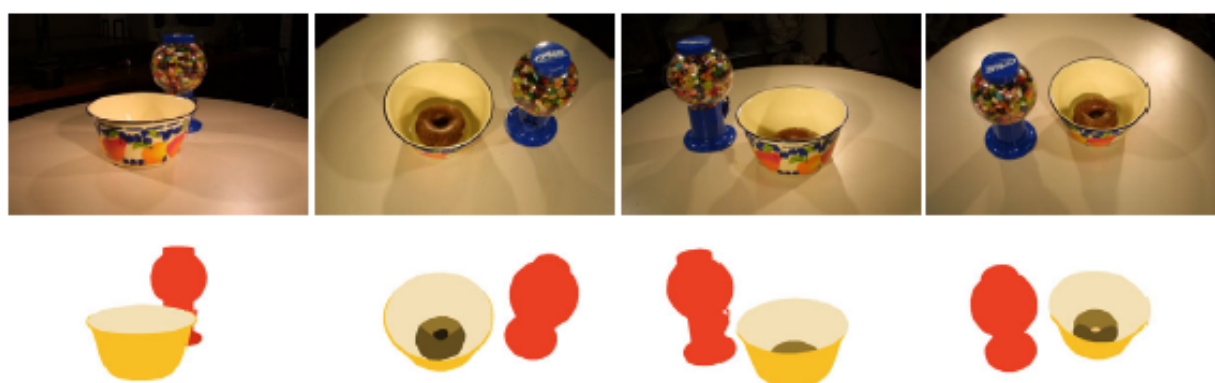


FIG. 7 – Quatre images de la scène du bol et du bagel, ainsi que la segmentation fournie par l'utilisateur.

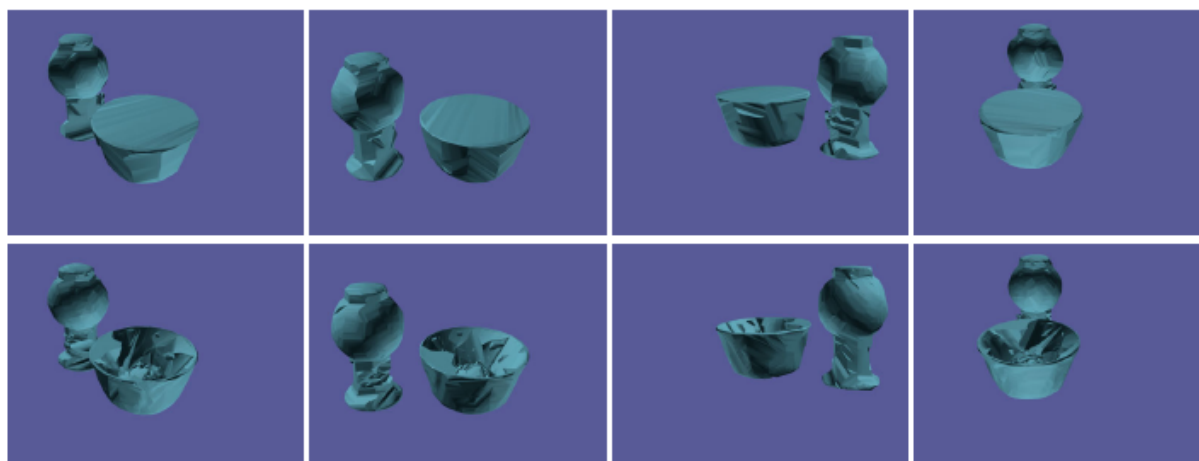


FIG. 8 – Quatre nouvelles vues de la scène reconstruite. Sur la rangée du haut, on voit l'enveloppe visuelle, et sur la rangée du bas, le résultat du Taraudage.

5 Conclusion

Dans le cadre de mon M2, je me suis placée dans la situation de l'article [15] : on considère un ensemble fini de photographies d'une même scène et l'objectif est de reconstituer le plus précisément possible l'enveloppe photo de la scène photographiée. On se place dans le cas où les photographies ont été précédemment segmentées. Cela revient à avoir identifié les silhouettes d'objets disjoints dans la scène. Cette segmentation permet de découper l'espace en cellules. Comme dans la méthode du Taraudage, on souhaite évider l'espace cellule après cellule pour converger vers l'enveloppe photo. Dans ce but, j'ai démontré rigoureusement le théorème qui permet de justifier que les cellules que l'on supprime à chaque itération sont bien "vides".

De nombreuses pistes semblent maintenant prometteuses à explorer. Il serait tout d'abord intéressant de prendre en compte les contraintes de tangence des objets dans les cellules. De plus, il est clair que les cellules ne sont pas indépendantes les unes de autres. On peut donc envisager d'utiliser les méthodes de propagation de contraintes dans notre problème. Enfin, il faut trouver une manière efficace d'implémenter cet algorithme.

Références

- [1] Bruce G. Baumgart, *A polyhedron representation for computer vision*, dans AFIPS National Computer Conference, 1975.
- [2] Aldo Laurentini, *The Visual Hull Concept for Silhouette Based Image Understanding*, IEEE PAMI, 16, 2, pp. 150-162, 1994.
- [3] Gregory Slabaugh, Bruce Culbertson, Thomas Malzbender et Ronald Scafe, *A Survey on Methods for Volumetric Scene Reconstruction from Photographs*, dans International Workshop on Volume Graphics, 2001.
- [4] Charles R. Dyer, *Volumetric Scene Reconstruction from Multiple Views*, dans L.S. Davis, editor, *Foundations of Image Understanding*, pages 469-489, Kluwer, Boston, 2001.
- [5] Bernard Chazelle, *An Optimal Algorithm for Intersecting Three-Dimensional Convex Polyhedra*, dans SIAM J. Computing, 21, pp. 671-696, 1992.
- [6] Ari Rappoport et Steven Spitz, *Interactive Boolean Operations for Conceptual Design of 3D Solids*, dans SIGGRAPH, pp 269-278, 1997.
- [7] Wojciech Matusik, Chris Buehler et Leonard McMillan, *Polyhedral Visual Hulls for Real-Time Rendering*, Eurographics Workshop on Rendering 2001.
- [8] Edmond Boyer et Jean-Sébastien Franco, *A hybrid approach for computing visual hulls of complex objects*, dans CVPR, volume I, pages 695-701, 2003.
- [9] Jean-Sébastien Franco et Edmond Boyer, *Exact Polyhedral Visual Hulls*, dans British Machine Vision Conference 2003.
- [10] Jean-Sébastien Franco et Edmond Boyer, *Efficient Polyhedral Modeling from Silhouettes*, dans IEEE transactions on PAMI, 31(3) :414-427, March 2009.
- [11] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Leonard McMillan et Steven Gortler, *Image-Based Visual Hulls*, dans Proceedings of SIGGRAPH, 2000.
- [12] Svetlana Lazebnik, Yasutaka Furukawa et Jean Ponce, *Projective Visual Hulls*, dans International Journal of Computer Vision, August 2007.
- [13] Kiriakos N. Kutulakos et Steven M. Seitz, *A Theory of Shape by Space Carving*, International Journal of Computer Vision, vol. 38, no. 3, pp. 199-218, 2000.
- [14] Diego Rother et Guillermo Sapiro, *Seeing 3D Objects in a Single 2D Image*, Accept pour publication à Proceedings of the 12th IEEE International Conference on Computer Vision, 2009.
- [15] Remo Ziegler, Wojciech Matusik, Hanspeter Pfister et Leonard McMillan, *3D Reconstruction Using Labeled Image Regions*, Symposium on Geometry Processing, 2003.
- [16] David A. Forsyth et Jean Ponce, *Computer Vision : A Modern Approach*, Prentice-Hall, 2003.
- [17] Richard Hartley et Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, June 2000.
- [18] Olivier Faugeras, *Three-Dimensional Computer Vision : A Geometric Viewpoint*, ser. Artificial Intelligence. MIT Press, Cambridge, 1993.