Amélioration d'une reconstruction 3D par voxels ('Visual Hull') à l'aide de la stéréo-vision

Eric Nowak – DEA IARFA – 2002/2003

Rapport de stage de DEA
Lieu du stage : INRIA de Rocquencourt
Projet : MIRAGES
Responsable de stage : Philippe Gérard

Résumé

Ce rapport propose une méthode innovante de construction d'enveloppes corporelles (maillages) d'un objet 3D à partir de photos. Dans un premier temps, on calcule le 'Visual Hull' de cet objet : une intersection volumique des projections de ses silhouettes. Le 'Visual Hull' induit des erreurs de modélisation des zones concaves. On va réduire ces erreurs grâce à une méthode de stéréo-vision : VHBS, *visual hull based stereo*. Cette dernière utilise le 'Visual Hull' dans deux buts : déterminer une zone de recherche sur la droite épipolaire, et effectuer un *camera mapping* permettant de comparer correctement des motifs issus de 2 vues. Les informations fournies par VHBS sont ensuite utilisées pour déformer le maillage 3D à l'aide de RBF.

Mots clés

[Visual Hull] [Stereo vision] [Image-based modeling] [Mesh deformation]

.

Je tiens à remercier sincèrement mes collègues du bâtiment 18 de l'INRIA de Rocquencourt grâce à qui mon stage s'est déroulé dans les meilleures conditions.
Je pense à l'assistante de projet, aux thésards et stagiaires des projets MERLIN, AXIS, et bien évidemment MIRAGES. Je remercie tout particulièrement Philippe Gérard, avec qui j'ai travaillé en étroite collaboration, pour ses conseils et son investissement. Enfin, je remercie André Gagalowicz pour la confiance qu'il m'a accordée en m'accueillant au sein de son équipe.

Sommaire

Illustrations

Les illustrations sont regroupées en annexe, pages 20 à 31

1 - Le projet global , l'objectif précis du stage

Le stage de DEA effectué entre avril et septembre 2003 s'inscrit dans un projet RIAM de 33 mois nommé *Golf-Stream*. Ce projet réunit l'équipe Mirages (Manipulation d'objets dans les Images pour la Réalité Augmentée et la Génération d' Effets Spéciaux) de l'INRIA de Rocquencourt, SYMAH VISION, la PGA (Professional Golfers' Association) et la FFG (Fédération Française de Golf).

Il s'agit de réaliser un outil de capture de mouvements et de réalité augmentée dédié au golf. Les applications sont multiples : les informations calculées et surincrustées à la vidéo originale permettent d'affiner la pédagogie sportive et de présenter de nouvelles images enrichies aux téléspectateurs et aux commentateurs sportifs. Ce projet de recherche comprend également une approche anthropométrique et biomécanique de la modélisation humaine.
Pour cela, les joueurs sont filmés par un ensemble de caméras synchronisées, les mouvements 3D des joueurs sont capturés puis enrichis (ou *augmentés*).
Le rôle de l'INRIA dans le projet est de réaliser cette capture du mouvement 3D des joueurs, sans utiliser de marqueurs qui pourraient gêner la pratique sportive (physiquement ou psychologiquement). On qualifie ce tracking 3D de *non intrusif*.

Pour des raisons précisées en annexe[1], ce tracking s'effectue avec un modèle 3D.

Le but du stage est de réaliser en partie ce modèle. En effet, le squelette du modèle existant déjà, il faut créer l'enveloppe du corps humain.
Plus précisément, il s'agit de modéliser l'enveloppe corporelle d'un joueur, photographié par des caméras synchronisées (sept dans notre cas). Contrainte imposée : réduire au strict minimum les interventions humaines lors de la modélisation : le processus doit être le plus automatisé possible.

Deux approches opposées ont été adoptées par le laboratoire de l'INRIA :

- approche contours : retrouver directement la surface du joueur en se basant sur les *contours* des silhouettes du joueur. Cette approche a été choisie par un collègue.
- approche volume : retrouver la surface du joueur après avoir estimé son volume, en se basant sur *l'intérieur* des silhouettes. C'est l'approche présentée dans ce document.

Nous présenterons les méthodes utilisées par la communauté scientifique, et expliquerons la démarche adoptée et la motivation des choix effectués. Cela nous mènera à l' exposé d'une méthode innovante de calculs de correspondances par stéréo-vision à base de 'Visual Hull' (*VHBS*). Nous détaillerons ensuite les étapes du processus de création du maillage. Après avoir présenté les résultats obtenus, nous conclurons en mentionnant l'orientation des prochains travaux.

2 - Comment y parvenir : démarche et état de l'art

2.1 - Types de modèles généralement utilisés

On présente tout d'abord les avantages et les inconvénients des techniques utilisées pour obtenir un modèle correspondant à l'objet à poursuivre, ou *tracker*. De plus amples détails sur le tracking non intrusif à base de modèles 3D et les modèles 3D utilisés sont fournis dans la bibliographie en annexe[1].

2.1.1 - Techniques nécessitant une forte interaction avec l'utilisateur

Les techniques suivantes sont utilisées par la communauté scientifique mais ne conviennent pas à notre projet, car elles manquent de robustesse et imposent une trop grande interaction utilisateur (généralement un infographiste).

[1] 8.4 - , *Bibliographie sur le tracking,*

- Façonnage manuel : travail fastidieux effectué par un infographiste. Le résultat est précis et esthétique, mais il est entièrement manuel.

- Modèles paramétriques [Lewis 00] : ce sont des modèles qui se modifient à l'aide de paramètres telle que : la taille, le poids, la cambrure du dos, le genre, l'importance de la musculation, etc. Ces modèles requièrent un paramétrage difficile et manuel, de plus, ils sont trop génériques pour représenter fidèlement un corps humain en particulier.

- Modification d'un modèle existant par un maillage de contrôle [Sun 99]. On peut définir un lien hiérarchique entre un maillage de très haute résolution et un maillage de faible résolution. Ainsi, des déformations appliquées au maillage de faible résolution induisent directement des déformations au maillage de plus haute résolution. Cette méthode est intéressante si les points du maillage basse résolution sont des points caractéristiques (extrémité du coude, menton, nez, genou, etc.). Ces points sont faciles à placer manuellement, mais il est difficile de les placer automatiquement.

- Déformations à partir de points caractéristiques placés dans l'espace. Même principe que précédemment, hormis que les déformations affectent directement un maillage de haute résolution. Mêmes problèmes que ci-dessus, auxquels s'ajoute celui du pré-positionnement du maillage, sans quoi les déformations donneraient des résultats aberrants.

2.1.2 - Techniques nécessitant peu d'interaction avec l'utilisateur

- Scanners : ils sont très précis et fournissent des nuages de points 3D très denses. Cependant, ils sont coûteux et requièrent une extrême immobilité d'un corps humain lors de la durée des mesures (de une à plusieurs dizaines de secondes).

- Ajustement de modèles [Hilton 00] : les points caractéristiques sont repérés automatiquement dans la 2D, et des correspondances entre les caractéristiques des images permettent de modifier le maillage dans l'espace. Cette technique nécessite l'utilisations de vues orthogonales, et impose au sujet de prendre la même posture que celle du modèle.

- Approche ellipse / contour : [Weik 00] . Des ellipses sont placées le long d'un squelette, donné ou estimé. Ces ellipses sont ensuite reliées pour créer une surface. Mon collègue utilise deux vues pour créer les ellipses, et les autres pour les déformer. Elles sont ensuite reliées par des surfaces.

- Visual Hull. *Détails en 2.3.2 - et 8.4* - . Les parties convexes sont correctement représentées, mais les parties concaves contiennent de grandes imprécisions [Slabaugh 01], [Cheung 00] , [Moezzi 00]

2.2 - Un tracking optimal

Il faut garder à l'esprit l'utilisation qui sera faite du modèle : il s'agit d'effectuer un tracking. La méthode de tracking choisie, mise en œuvre par le laboratoire, est la suivante :

- Placer la marionnette 3D ajustée morphologiquement dans la même posture que le joueur sur la première image de la séquence vidéo à analyser.
- Apprendre la texture de l'objet.
- Tracker les mouvements d'un humain en tentant de le mimer à l'aide du clone synthétique.

Le modèle doit donc permettre d'effectuer un tracking général, et ce type de tracking en particulier. Une bibliographie a été réalisée sur le tracking à ce propos, elle est présentée en annexe[2]. La littérature met l'accent sur l'importance du modèle, de sa précision et des contraintes qu'il impose.

[2] *8.4 - Bibliographie sur le tracking,*

Il faut donc disposer :

- d'un squelette morphologiquement adapté au personnage filmé et dont les degrés de liberté sont contraints: l'élaboration d'un tel squelette est l'objet de travaux dans le laboratoire
- d'une enveloppe précise, de bonne résolution, pour que le découpage et le plaquage de texture ne donne pas d'aberrations

Il faut noter qu'une telle démarche est inutile si l'adaptation du squelette dans l'enveloppe corporelle s'avère impossible : disposer d'une enveloppe et d'un squelette, aussi parfaits soient-ils, est inutile si le squelette ne peut être positionné correctement dans l'enveloppe, et si le skinning[3] ne peut être déterminé automatiquement. Cette difficulté explique pourquoi la littérature mentionne de nombreuses approches visant à déformer des enveloppes corporelles où les squelettes sont déjà positionnés. Une méthode d'intégration du squelette et de skinning automatique a donc été proposée (mais n'a pas été testée). Comme elle semble réalisable[4] le calcul de l'enveloppe corporelle peut être effectué.

2.3 - Un 'Visual Hull' optimal

Le paragraphe suivant présente des informations concernant les 'Visual Hulls' indispensables à la compréhension de ce document. Pour plus de détails, consulter [Hilton 00] ou le paragraphe 8.4 - .

2.3.1 - Définition et création d'un 'Visual Hull'

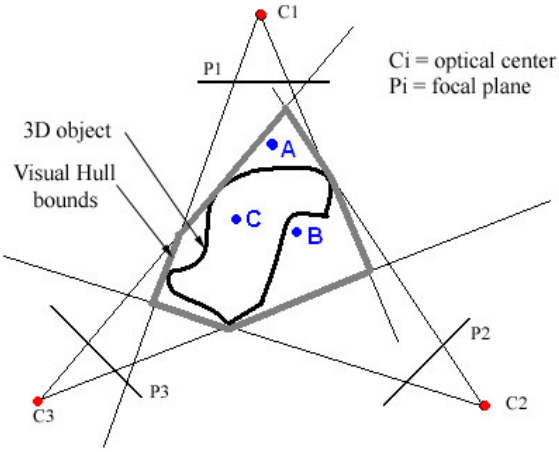


Figure 1 – Principe du Visual Hull

Considérons un objet positionné dans l'espace, photographié par *n* appareils-photo simultanément. On peut alors définir *n* silhouettes[5] différentes de cet objet.

Si un point est inclus dans le volume de l'objet, il se projète dans toutes les silhouettes (par définition). Mais la réciproque est fausse : un point qui se projète dans toutes les silhouettes n'est pas forcément un point de l'objet. *Figure 1* : le point C fait partie de l'objet, il se projète donc dans toutes les silhouettes. Les points A et B ne font pas partie de l'objet, ils se projètent pourtant dans toutes les silhouettes.

On définit le *'Visual Hull' (VH)* comme l'ensemble des points 3D de l'espace qui se projètent dans toutes les silhouettes. Ainsi, les points A, B et C de la *Figure 1* sont des points du VH. Etant donné ce qui a été dit plus haut, le VH contient l'objet qu'il modélise.

Le VH peut être créé à partir de voxels : des cubes élémentaires. Un cube fait partie du VH si, et seulement si, il se projète intégralement dans toutes les silhouettes de l'objet à modéliser. On obtient le VH par l'union de tous ces cubes.

Exemples de VH :
Figure 2 : un VH de faible résolution, et en arrière plan la photo de l'objet 3D à modéliser
Figure 3 : un VH de haute résolution

[3] Skinning : déplacement des sommets du maillage en fonction des mouvements des os du squelette
[4] Méthode explicitée en 8.2.1 -
[5] Silhouette : ensemble des pixels situés à l'intérieur de la projection 2D (image) d'un objet 3D. Les objets étudiés étant connexes, les silhouettes le sont donc aussi.

2.3.2 - Limitations

Notons *VH(n)* le *Visual Hull* réalisé avec *n* vues différentes et *VO* l'objet à modéliser, considéré en tant que volume.

Caractéristiques importantes, démontrées par [Laurentini 94] :

- plus le nombre de caméras augmente, plus l'écart entre VH(n) et VO se réduit :
  Le volume V(n) = Volume(VH(n) \ VO) est décroissant en n
- Pour certains objets à modéliser (ceux présentant des concavités) :
  $\lim_{n=+\infty} (V(n)) = k > 0$

Bilan :

- Il faut avoir suffisamment de caméras bien disposées pour réduire l'erreur à une valeur proche du minimum. *Figure 1* : le point A peut être éliminé du VH à l'aide d'une vue horizontale placée à la hauteur du VH
- Il faut renoncer à modéliser les parties convexes avec un VH. *Figure 1* : le point B ne pourra jamais être éliminé du VH en n'utilisant que les silhouettes

2.3.3 - Au-delà de ces limitations

La notion de 'Hull Carving'[6] permet de creuser le VH, de manière à éliminer les voxels des parties concaves. Cette méthode ne nous convient pas : le 'Hull Carving' est basé sur l'observation de la couleur d'un seul voxel, et non pas sur un ensemble de voxels définissant un voisinage. En effet, les couleurs des voxels du voisinage ne peuvent être considérées car l'algorithme du 'Hull Carving' n'a pas forcément validé ces voxels. Il y a donc des risques de faux appariements importants.

Une méthode permet de prendre en compte les pixels et leur voisinage lors du calcul des appariements : la stéréo-vision[7]. Cette méthode a cependant deux défauts majeurs. Premièrement, elle se montre peu efficace dans notre cadre : les distances inter-cameras sont élevées et les directions des caméras sont concourantes. Deuxièmement, les motifs comparés d'une caméra à l'autre ne sont pas forcément de même taille et de même orientation en raison des paramètres intrinsèques et extrinsèques des caméras. Illustration *Figure 6* : le motif orange[8] vu en C_1 est très déformé en C_2 et en C_3, le calcul des correspondances s'avère difficile (voir impossible dans un environnement plus complexe).

Aussi faut-il définir une méthode d'appariement en stéréo-vision ne présentant pas ces inconvénients.

2.4 - Une stéréo-vision optimale

Des précisions sur la stéréo-vision classique et le calcul des correspondances se trouvent en annexe[7]

[Debevec 96], dans le but de modéliser finement des monuments architecturaux, propose une méthode de stéréo-vision épipolaire permettant de s'affranchir de ces problèmes. *Figure 8 :* au lieu d'apparier directement les points p_k et p_o, très distants, il effectue un warping lui permettant de réduire les disparités :

1) Un modèle simple (*approximate structure*) de l'objet à modéliser (*actual structure*) est construit manuellement : les éléments du modèle sont des surfaces rectangulaires représentant des pans de murs (*Figure 8*)
2) Deux photos sont prises, l'une avec le point de vue *Key*, l'autre avec le point de vue *Offset* (*Figure 7a*, *Figure 7c*)

[6] *8.4 - Bibliographie sur le tracking,*
[7] Rappels : 8.3.1 - *Géométrie épipolaire* et 8.3.2 - *Calcul des correspondances*
[8] Rectangulaire et tacheté (remarque pour la version noir et blanc du rapport)

3) L'image *Offset* est projetée sur le modèle, et un rendu texturé de ce modèle est effectué depuis le point de vue *Key* : on obtient une image nommée *Warped*[9] *Offset (Figure 7b)*. Comme le modèle présente peu de différences par rapport à l'objet réel, l'image du modèle texturé (*Warped offset*) est donc proche de l'image de l'objet réel (*Key*) (*Figure 7*a et *Figure 7b*).
4) L'auteur cherche ensuite à apparier les points p_k et $q_k$en parcourant la droite épipolaire e_k sur une faible distance[10] (*Figure 8*).

Le concept de warping nous permettra d'effectuer des appariements stéréo corrects grâce au VH. C'est l'objet du chapitre suivant

3 - Nouvelle méthode de stéréo-vision : VHBS

Précisons tout d'abord le terme de *camera mapping*. Etant donnés un maillage et une caméra perspective, on définit par *camera mapping* l'opération consistant à projeter l'ensemble des sommets du maillage sur le plan film de la caméra dans le but de découper la texture dans l'image qu'elle a filmée. *Figure 9* : (a) : M vu par C sans texture. (b) image I (c) M vu par C après *camera mapping*. (d) M vu par une autre caméra après *camera mapping*
On notera MBS la méthode *Model Based Stereo* de [Debevec 96] et VHBS la méthode que nous proposons : *Visual Hull Based Stereo*.

3.1 - Analyse des différences : 'MBS vs *VHBS'*

3.1.1 - Invisibilité

Le *camera mapping* est une opération qui nous permet de rendre deux images prises de deux vues différentes comparables. En effet, si on projète la photo d'un objet 3D sur un modèle proche de celui-ci, on obtient un maillage texturé identique (aux défauts de modélisation près) à l'objet 3D initial.
Dans le cadre de MBS, le camera mapping est effectué sur une surface plane : il s'agit donc d'un simple warping. L'image *Offset* est donc transformée en une image *Warped offset* très similaire à l'image *Key*.

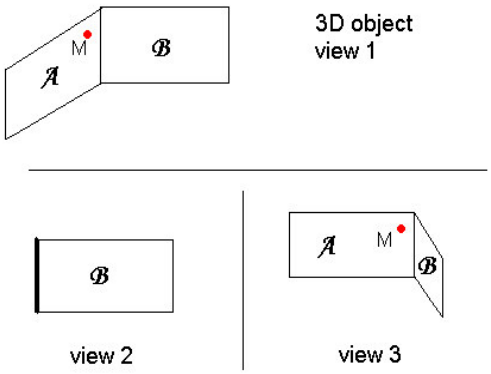


Figure 10 – Problèmes de visibilité du point M

Dans notre cas, le camera mapping est effectué sur un très grand nombre de faces, qui n'ont pas toutes la même orientation. L'image n'est donc pas déformée globalement, mais chaque face subit une déformation particulière. De plus, le fait d'effectuer un mapping sur un volume pose un problème de visibilité.
Figure 10 : Supposons que l'on effectue un *camera mapping* sur l'objet 3D présenté dans la vue 1.
On effectue le mapping dans la vue 2 : la face B, visible, est alors texturée, mais la face A, invisible, ne l'est pas. Si on observe l'objet après *camera mapping* dans la vue 3, la partie B sera visible (car texturée) mais la partie A sera invisible. Le passage de la vue 2 à la vue 3 est donc uniquement exploitable pour une seule des 2 faces, la face B. En effet, étant donné le point M de la face A visible dans la vue 3, si on cherche à l'apparier à un point de la vue 2, un point de la face B sera choisi, car la face A est invisible. Il y a donc inévitablement une erreur d'appariement.

3.1.2 - Erreurs de modélisations

Les erreurs de modélisation de l'objet 3D déterminent la ressemblance entre les images *Key* et *Warped offset*. Plus le modèle est précis, plus les images se ressemblent. Dans le cas de MBS, les faces de

[9] Projeter une image sur une surface et l'observer depuis un point de vue où celle-ci reste visible revient à effectuer un warping élémentaire.
[10] « faible distance » n'est pas quantifié dans l'article.

l'objet à modéliser sont presque planes, il y a donc peu d'erreurs, ce qui explique pourquoi les disparités sont faibles. Dans notre cas, la principe du 'Visual Hull' induit d'importantes erreurs de modélisation des parties concaves. Les erreurs de modélisation peuvent alors être importantes, ce qui implique que les disparités entre les images *Key* et *Warped offset* le sont aussi.

Il y a alors un dilemme dans la recherche épipolaire. On peut chercher les correspondances sur un zone étendue (augmente les chances de faux appariements, assure de passer par le point correct) ou une zone réduite (diminue les chances de faux appariements, mais le point cherché est peut être hors de la zone). Ce problème vient du fait que la recherche épipolaire fournit toujours un meilleur candidat : même si aucun point inspecté ne correspond au point cherché, le meilleur parmi ceux-ci est sélectionné.

3.2 - Résolution du problème du manque d'information

Afin de résoudre le premier problème présenté, nous introduisons le concept de visibilité.
Il permet de différencier une information sure (une face visible est correctement texturée) d'une information erronée (une face invisible est mal texturée).

Etant données les images *Key (K)*, *Offset (O)*, et *Warped offset (WO)*, la méthode MBS utilise un point de *Key* et cherche des correspondances dans *Warped offset (Figure 8)*. L'appariement est fait avec le point offrant la valeur de correspondance minimale. Dans notre cas, il faut s'assurer de la validité de l'information fournie par l'image *Warped offset* : si on cherche dans *Warped offset* le point correspondant à un point de *Key*, il faut que les points inspectés dans *Warped offset* et les points de leurs voisinages soient visibles.

Notons $I(i,j)$ le niveau de gris de l'image *Key*, et $I'(i,j)$ le niveau de gris de l'image *Warped Offset,* et considérons l'exemple du calcul de correspondances avec la méthode SAD : *sum of absolute difference.* La valeur de correspondance entre $P=(i,j)\in K$ et $P'=(k,l)\in WO$ est :

$$\Delta(P,P')=\sum_{u=-hs}^{hs}\sum_{v=-hs}^{hs}\left|I(i+u,j+v)-I'(k+u,l+v)\right|$$

où $(2\,hs+1)^2$ est la taille de la fenêtre de calcul de correspondances.

Si $I'(k+u, l+v)$ n'est pas défini (face invisible) le calcul n'a plus de sens. De plus, cette valeur ne peut être ignorée dans la somme : dans ce cas, un point P' situé dans une zone totalement invisible aurait une valeur de correspondance nulle (aucun terme dans la somme) et serait choisi comme meilleur appariement.

Il faut donc procéder différemment, et choisir un point P' de *Warped offset* dans une zone visible, et chercher les correspondances dans *Key*, où les seuls pixels invisibles se situent dans l'arrière-plan de l'image. En ces points, la valeur $I(i+u, j+v)$ n'est donc pas définie.
Figure 11 : I_1 représente l'image *Key*, I2 l'image *Warped Offset*. La zone blanche est une face du modèle texturée par *camera mapping*. Le point P1 est invisible dans l'image I1 (dans l' arrière-plan) mais visible dans l'image I2 (car il est sur une face du modèle). Les deux points P1 se correspondant, il faut avoir : $I(P1) = I'(P1)$ pour apparier correctement.

Ainsi, si *b* est la couleur du fond d'écran utilisé pendant le camera mapping, on redéfinit la fonction de correspondance comme :

$$\Delta\left(P=(i,j),P'=(k,l)\right)=\sum_{\substack{u,v \text{ dans } [-hs,hs]^2,\\ (k+u,l+v) \text{ visible dans WO}}}\left|\tilde{I}(i+u,j+v)-I'(k+u,l+v)\right| \qquad (1)$$

$$\text{où } \tilde{I}(\alpha,\beta)=\begin{cases} I(\alpha,\beta) \text{ si } (\alpha,\beta) \text{ visible dans K} \\ b \text{ sinon} \end{cases}$$

Conclusion :

Pour apparier deux points *P* et *P'* par stéréo-vision il faut :

- calculer la visibilité des faces du maillage dans l'image *Warped Offset*
- choisir une fenêtre visible dans l'image *Warped Offset,* et définir comme *P'* le pixel central
- déterminer le pixel *P* de l'image *Key* qui minimise l'équation (1)

Le chapitre *8.2.2* - de l'annexe apporte des précisions sur le calcul de la visibilité des faces.

3.3 - Résolution du dilemme dû à l'étendue de la zone de recherche

Si le modèle utilisé est parfait (identique à l'objet à modéliser), les images *Key* et *Warped offset* sont identiques. Il n'y a pas de disparités entre $P \in K$ et $P' \in WO$ si P et P' sont deux points correspondants. Moins le modèle est parfait, plus la position du point P est éloignée de celle de P'. Dans la méthode MBS, ces points sont très proches, car le modèle est très semblable à la réalité.

Observons ci-dessous le schéma fondamental du VHBS (*Figure 12*), indispensable à la compréhension de ce qui suit. Il se trouve en plus grande dimension en annexe, page 24.

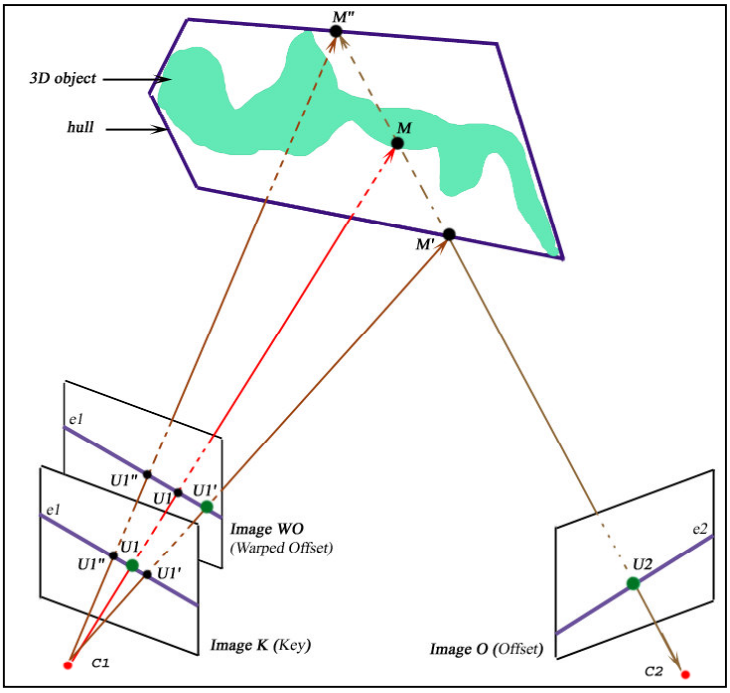


Figure 12 – Principe VHBS : schéma fondamental

On dispose de deux caméras C_1 et C_2, d'un objet 3D à modéliser, et le 'Visual Hull' (VH) de cet objet a été calculé. Deux photos ont été prises avec C_1 et C_2. On obtient respectivement les images *Key* et *Offset*.

Le point M est vu dans *Key* en U_1 et dans *Offset* en U_2. Le but de la stéréo épipolaire classique est d'associer les points U_1 et U_2 pour retrouver M par intersection des rayons optiques (C_1, U_1) et (C_2, U_2).

Dans notre cas, l'image *Offset* est projetée sur le VH, puis elle est observée depuis le point de vue C_1 : on obtient l'image *Warped offset* (positionnée devant l'image *Key* pour que le schéma soit plus lisible, en réalité elle sont superposées, par définition du camera mapping).

Le point U_2 étant projeté en M' lors du camera mapping, il est vu dans *Warped offset* en U_1', point d'intersection entre le plan focal de la caméra C_1 et le rayon optique (C_1, M'). Voici l'élément clé de cette méthode : du fait du camera mapping, le point U_1 dans *Key* et son voisinage sont semblables au point U_1' dans *Warped offset* et à son voisinage.

Le nouvel enjeu n'est donc plus d'associer les points U_1 et U_2 (U_2 est sur la droite épipolaire e_2, mais on ne dispose pas de plus d'informations) mais les points U_1 et U_1'. En effet, connaissant U_1', on peut déduire la position du point M' (par projection sur le VH) puis celle du point U_2 (par projection dans le plan focal de la caméra C_2), puis celle du point M, par intersection des rayons optiques (C_1, U_1) et (C_2, U_2).

Ayant choisi un point U_1' dans *Warped offset*, il faut donc déterminer le point U_1 correspondant dans *Key*.

Le point M est situé dans le VH, car celui-ci contient l'objet 3D. C'est là tout l'intérêt d'utiliser un VH plutôt qu'un autre modèle approximatif : le point M se trouve *obligatoirement* dans le VH. En particulier, il se situe sur le rayon optique (C_2, U_2) car U_2 est la projection de M dans *Offset*. Le point M est donc situé sur le segment [M', M''].[11]

Si le VH était parfait, on aurait : M=M'. Moins le VH est précis, plus la distance d(M, M') est élevée, et plus il y aura de correspondances à calculer. Dans quelle région chercher U_1 ? M étant situé sur [M', M''], l'image de M est située sur l'image de [M', M''] : il faut donc effectuer les recherches de correspondances sur un segment de la droite épipolaire : le segment [U_1', U_1''], où l'on définit U_1'' comme l'intersection du plan focal de la camera C_1 et du rayon optique (C_1, M'').

Bilan :
- Il faut apparier U_1 et U_1'
- U_1 se situe toujours entre U_1' et U_1''
- si le VH est parfait, M = M' donc U_1 = U_1'
- moins le VH est parfait, plus U_1 s'éloigne de U_1', pour se rapprocher de U_1''.

Cela nous permet de définir deux critères pour chercher le point U_1 :

1) Zone de présence certaine : U_1 est forcément situé sur le segment [U_1', U_1'']. Nous avons donc réduit la zone de recherche de U_1 de la droite épipolaire (U_1', U_1'') au segment épipolaire [U1', U1'']. Ce segment est défini par la partie du rayon optique (C_2, M') qui traverse le VH. Si le segment est court, la zone de recherche est réduite, les appariements sont faciles à effectuer. Dans ce cas, le dilemme est résolu. Mais si le segment est long, une grande zone de l'image est inspectée, ce qui augmente les chances de faux appariements. Nous pouvons donc utiliser un second critère :

2) Disparité maximale estimée : on peut imposer à l'algorithme de chercher U_1 sur le segment [U_1', U_1''], en commençant par U_1' (VH parfait U_1'= U_1) sans inspecter plus de *x* pixels. Cette valeur x est une limite arbitraire qui ne peut être calculée, mais qui peut être estimée par un humain. Pour garder l'humain hors du processus, il suffit d'affecter à *x* une valeur élevée (supérieure à la longueur de la diagonale de l'image). Mais si un ordre de grandeur de *x* peut être estimé, les résultats sont améliorés. *Figure 28* : les disparités sont faibles entre les images *Key* et *Warped offset*. La disparité visuelle maximale peut aisément être estimée inférieure à 10 pixels.

3.4 - Algorithme de stéréo-vision proposé (VHBS)

Les notations font référence à la *Figure 12.*

Les différentes étapes de l'algorithme complet du *Visual Hull Based Stereo* (VHBS) sont détaillées ci-dessous. Cet algorithme permet de déterminer la position 3D d'un point d'un objet à modéliser, dont on dispose du 'Visual Hull' (VH). Deux prises de vue de cet objet, *Image Key* (*K*) et *Image Offset* (*O*), sont respectivement réalisées par les caméras C_1 et C_2.

Camera mapping du maillage du VH par la caméra C_2 (qui génère l'image *Image Offset*)
Cette étape applique une texture à chaque face du maillage.
Les sommets qui composent la face F_i sont F_{i1}, F_{i2} et F_{i3}. Ces sommets se projètent dans l'image *Offset* en P_{i1}, P_{i2} et P_{i3}. On affecte alors aux sommets F_{ij} de la face F_i les coordonnées de texture P_{ij}. Tout ce passe comme si la face « découpait » sa texture dans l'image *Offset* à la position où elle se trouve.

[11] [Li 02] estime lui aussi la position du point M, mais n'utilise pas de camera mapping pour calculer les correspondances, il a donc plus de difficultés à apparier correctement

Comme le VH contient l'objet à modéliser, la texture est intégralement plaquée sur le VH. Ainsi, l'image obtenue en effectuant un rendu du VH texturé dans la vue C_2 est rigoureusement égale à l'image *Offset*. On construit l'image *Warped Offset* par un rendu de VH dans la vue C_1.

Sélection d'un point caractéristique.
On choisit un point U_1' parmi la liste des points caractéristiques détectés dans l'image[12].

Calcul de la position 3D de M'
M' est le projeté de U_1' sur le VH : c'est la première intersection du maillage et du rayon optique de direction (C_1, U_1'), d'origine C_1

Calcul de la position 2D de U_2
U_2 est le projeté de M' dans l'image *Offset.* U_2 se calcule par :

$$[\alpha \ \beta \ \gamma]^t = \mathbf{W2P}_{C2} \cdot \mathbf{M'}$$ [13] et $$U_2 = \left(\frac{\alpha}{\gamma}, \frac{\beta}{\gamma} \right)$$

Calcul de la position 3D de M''
M'' est le dernier point d'intersection du maillage et du rayon optique de direction (C_1, U_1'), d'origine C_1.

Calcul de la position 2D de U_1''
U_1'' est le projeté de M'' dans *Warped Offset*.
U_1'' se calcule par :

$$[\alpha \ \beta \ \gamma]^t = \mathbf{W2P}_{C1} \cdot \mathbf{M''}$$ [13] et $$U_1'' = \left(\frac{\alpha}{\gamma}, \frac{\beta}{\gamma} \right)$$

Recherche de la position 2D de U_1
U_1 est le point à apparier avec U_1'. Il se trouve sur le segment [U_1', U_1''], qui est inclus dans la droite épipolaire e_1. Par soucis de précision (arrondis dû à un univers de travail discret (pixels)), le segment épipolaire est parcouru ainsi que le segment situé un pixel au-dessus et celui situé un pixel au-dessous. De plus, si une disparité visuelle maximum *x* est donnée, seuls les *x* premiers pixels de ces segments sont considérés (en commençant par U_1'). Soit Ω l'ensemble des pixels de ces trois segments. Il s'agit de l'ensemble des candidats à l'appariement avec U_1. On définit U_1 comme :

$$U_1 = \underset{U \in \Omega}{\text{Argmax}} \{ \Delta(U, U_1') \}$$

Calcul de la position 3D de M
La position 3D de M peut être déduite de U_1 et U_2 :

$$M = D_1 \cap D_2 \ , \quad D_1 = (C_1, U_1), \quad D_2 = (C_2, U_2)$$

En raison des approximations effectuées pendant les calculs (arrondis et discrétisation) les droites D_1 et D_2 ne se coupent pas réellement dans l'espace. On redéfinit donc M par 'line reconstruction' [Faugeras 93] :

$$M = \frac{M_1 + M_2}{2} \quad M_1 \in D_1 \quad M_2 \in D_2 \quad d(M_1, M_2) = d(D_1, D_2)$$

4 - Processus de création de l'enveloppe corporelle

La méthode de création volumique d'une enveloppe corporelle choisie fait appel à des théories issues de divers domaines. Le principe est de tirer parti des différentes approches : garder les avantages et compenser les inconvénients : le 'Visual Hull' (VH) permet d'obtenir une bonne modélisation des

[12] voir 4.4.1 - , *Détection des points caractéristiques*
[13] La matrice **W2P** est explicitée dans la partie Calibration

zones convexes mais supprime les zones concaves. La stéréo-vision positionne précisément des points dans l'espace, mais ne fournit pas un nuage suffisamment dense pour en déduire un maillage.

Les sous-parties suivantes présentent les différentes étapes du processus de création d'une enveloppe corporelle, dans l'ordre chronologique de leur déroulement.

4.1 - Calibration

Se référer à la *Figure 13.*

Une grille de calibration dont la géométrie est connue est placée de sorte qu'elle soit visible par toutes les caméras. Grâce à un outil de calibration avec correspondances 3D-2D , les positions des cameras, leurs orientations et leurs distances focales sont retrouvées (on suppose qu'il n'y a pas de distorsions, et que le centre optique se situe au centre de la caméra, modèle '*pinhole camera*').

On dispose des informations suivantes :

- Valeur du pixel ratio[14] : pixR (ici, norme vidéo : 1.1)
- Dimensions de l'image en pixels : L_p x H_p (ici : 720 x 576)
- Dimensions du plan film théorique en cm : L_c x H_c (ici : 3.6 x 2.4, format connu sous le nom *24x36*)

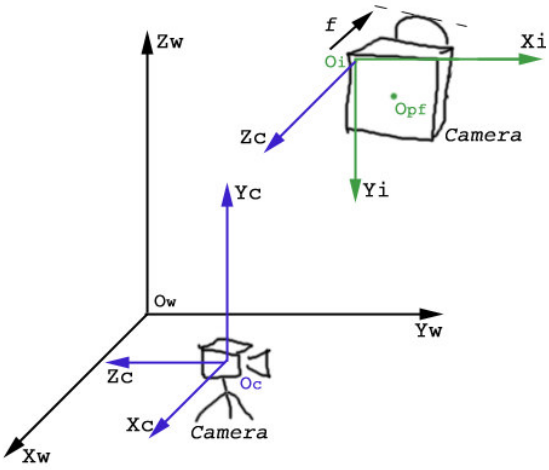
L'outil de calibration fournit :

- distance focale en cm : f_c
- position de la caméra en cm dans le repère monde : $\mathbf{T_c}$
- une matrice de rotation permettant de passer du repère monde au repère caméra : **R**

On définit :

- c_{pc} : coefficient de passage pixels → cm : $c_{pc}=\frac{Lc}{Lp}$
- f_p : distance focale en pixels, $f_p=\frac{f_c.Lp}{Lc}=\frac{f_c}{c_{pc}}$

Cela nous permet de calculer les formules de passage entre les repères significatifs de la scène :



Repère World : (O_w , x_w, y_w, z_w)
Repère Camera : (O_c , x_c, y_c, z_c)
Repère Image : (O_i , x_I, y_I)

1) <u>passage du repère monde au repère image</u> (matrice **W2P**, *world to pixels*). Si les coordonnées 3D d'un point dans le repère monde sont (x,y,z), ce point se projète dans l'image 2D en (U,V) défini par :

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \mathbf{W2P} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix};$$

[14] Rapport (hauteur d'un pixel) / (largeur d'un pixel), induisant une déformation de l'image

$$\mathbf{W2P}=\begin{bmatrix} 1 & 0 & \frac{H_p}{2} \\ 0 & 1 & \frac{L_p}{2} \\ 0 & 0 & 1 \end{bmatrix}\times\begin{bmatrix} -f_p & 0 & 0 \\ 0 & f_p\ .\ pixR & 0 \\ 0 & 0 & 1 \end{bmatrix}\times\begin{bmatrix} (\mathbf{R})_{3x3} & (-\mathbf{R}\times\mathbf{T_c})_{3x1} \end{bmatrix};$$

$$(U,V)=\left(\frac{u}{w},\frac{v}{w}\right) \text{ si } w\neq 0 \text{ (pas de projection dans l'image sinon)}$$

2) <u>passage du repère image au repère monde, dans le plan focal de la caméra considérée</u> (matrice **P2W**, *pixels to world*). Le point 3D (x,y,z) situé dans le plan focal de la caméra[15] et se projetant en (u,v) défini dans le repère image est le point :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}=\mathbf{P2W}\times\begin{bmatrix} u \\ v \\ 1 \end{bmatrix};$$

$$\mathbf{P2W}=\begin{bmatrix} (\mathbf{R'})_{3x3} & (\mathbf{T_c})_{3x1} \end{bmatrix}\times\begin{bmatrix} c_{pc} & 0 & 0 \\ 0 & -c_{pc} & 0 \\ 0 & 0 & -f_c \\ 0 & 0 & 1 \end{bmatrix}\times\begin{bmatrix} 1 & 0 & -\frac{H_p}{2} \\ 0 & 1 & -\frac{L_p}{2} \\ 0 & 0 & 1 \end{bmatrix}\times\begin{bmatrix} u \\ v \\ 1 \end{bmatrix};$$

Ces conversions se font à l'aide d'un simple produit matriciel une fois que les matrices **P2W** et **W2P** sont calculées. Les démonstrations sont en annexe, 8.2.3 - .

4.2 - Segmentation

La segmentation des images a été effectuée manuellement afin de traiter plus rapidement les autres aspects du problème.

4.3 - Création du 'Visual Hull'

4.3.1 - Création par octree

Nous avons opté pour la méthode de création de 'Visual Hull' (VH) par octree (*Figure 14*). Le principe est de tester si un cube se projète intégralement à l'intérieur de toutes les silhouettes de l'objet à modéliser. Si c'est le cas, le cube fait partie du VH. Sinon, si le cube se projète au moins une fois hors d'une silhouette, le cube est à l'extérieur du VH. Sinon, cela signifie que le cube se projète partiellement dans toutes les silhouettes. Le cube est alors divisé en 8 cubes partageant tous un sommet (le centre du cube initial), et le processus précédent est répété pour chacun des 8 cubes.

Cette méthode n'a d'intérêt que si la position d'un cube par rapport à la silhouette peut-être déterminée rapidement, sinon il est plus simple de travailler directement avec les voxels (cubes élémentaires).

[Szeliski 93] propose de créer un tableau de la taille de l'image, et dont les éléments (i , j) contiennent l'information suivante : « taille maximale du côté du carré entièrement à l'intérieur de la silhouette, et dont le coin inférieur droit est positionné en (i , j)» : *Figure 15.* Pour vérifier si un cube se projète à l'intérieur d'une silhouette, il considère le plus petit carré englobant de la projection des 8 sommets du cube, et observe la position relative du carré et de la silhouette grâce au tableau précédemment cité : *Figure 17*

[15] Dans un repère caméra classique dont l'axe des z serait dirigé vers l'arrière, la troisième coordonnée de ce point serait : - *f*. Dans le cas présent, ce sont les coordonnées dans le repère monde qui nous intéressent.

Cette méthode a le défaut d'utiliser des zones englobantes carrées, alors qu'en utilisant des zones englobantes rectangulaires les erreurs d'approximation de la surface de projection sont diminuées. En effet, en voulant remplacer un rectangle de taille l x L, $l \leq L$ par un carré de taille L^2, il y a une erreur d'approximation de valeur $L(L - l)$. Nous remplaçons donc les zones carrées par des zones rectangulaires, en découpant une zone rectangulaire en : (L / l) carrés si cette valeur est entière, $\lfloor L/l \rfloor + 1$ carrés sinon. Figure 29 : le rectangle de taille L x l a été modélisé par 3 carrés de côté l à gauche, et par 2 carrés de côté l à droite.

4.3.2 - Calcul de la résolution maximale du 'Visual Hull'

La résolution maximale du VH est la distance $\delta = \delta_{theorique}$ telle que la projection d'un cube de côté 2δ occupe plus d'un pixel dans au moins une vue, et la projection d'un cube de côté δ n'occupe qu'un pixel dans toutes les vues.
Soient L_c la largeur en cm du plan film théorique, et L_p la largeur en pixels des images. Soient Ci le centre optique de la caméra i, O_i le centre du plan focal physique de cette dernière et f_i sa distance focale. Enfin, soit Fi le point 3D de l'objet à modéliser le plus éloigné de la caméra i.

On définit alors $d_i = \frac{1}{\left\| \overrightarrow{C_i O_i} \right\|} \left| \left\langle \overrightarrow{C_i O_i}, \overrightarrow{C_i F_i} \right\rangle \right|$.

$$\text{Alors, } \delta_{theorique} = \min_{i=1..n\_cameras} \left\{ \frac{L_c}{L_p} \frac{d_i}{f_i} \right\}$$

La longueur du côté du plus petit cube de la scène est donc supérieure à cette valeur théorique.
Remarque : cette valeur théorique est calculée à l'aide d'une configuration de triangle de Thalès, les triangles pris en comptes ayant pour sommet commun C_i, pour première base le côté du cube le plus éloigné et pour seconde base la projection de ce côté, qui mesure un pixel.

4.3.3 - Transformation du 'Visual Hull' en maillage

L'algorithme utilisé pour transformer le volume en maillage est relativement simple. Définissons la connexité d'un voxel comme une 6-connexité : les voisins d'un voxel sont ses symétriques par rapport à chacune de ses six faces. L'algorithme de transformation parcourt tous les voxels composant le volume, et chaque fois qu'un voisin est absent, une face carrée est créée, identique à face frontière entre le voxel et l'emplacement de son voisin absent. Cette face est ensuite ajoutée au maillage. Cet algorithme produit un fort aliasing. Il est prévu d'utiliser l'algorithme du *marching cube* [Lorensen 87] dans les travaux ultérieurs afin de remédier à ce problème.

4.4 - Stéréo-vision et 'attracteurs'

4.4.1 - Détection des points caractéristiques

Afin d'effectuer des appariements, des points caractéristiques sont automatiquement extraits des images. Ces points doivent se différencier suffisamment de leurs voisins pour faciliter les appariements.
Si les niveaux de gris $NG(i , j)$ des pixels (i , j) sont compris entre 0 et 1, et que le seuil de différenciation minimal est f choisi arbitrairement tel que $0 \leq f \leq 1$, on définit un pixel (i , j) comme caractéristique si, et seulement si :

- $D_x = \max \left\{ \left| NG(i,j) - NG(i-1,j) \right|, \left| NG(i,j) - NG(i+1,j) \right| \right\} \geq f$:
  différenciation sur l'axe des abscisses

et

- $D_y = \max \left\{ \left| NG(i,j) - NG(i,j-1) \right|, \left| NG(i,j) - NG(i,j+1) \right| \right\} \geq f$:
  différenciation sur l'axe des ordonnées

De plus, afin de ne pas sélectionner trop de points dans un voisinage réduit, on choisit le pixel ayant la plus grande valeur de différenciation : *min* (D_x, D_y) dans une fenêtre de taille fixée.

4.4.2 - Calcul de la position 3D par stéréo-vision

Cette partie a fait l'objet du chapitre 3 -

4.4.3 - Choix du point du maillage à attirer

Le but de l'algorithme VHBS est de fournir un point *R* situé sur l'objet à modéliser. Le VH doit ensuite être déformé pour passer par *R*. Mais quel point du VH doit être déplacé pour atteindre *R* ? Quand l'algorithme VHBS est nécessaire (zone concave), choisir le point de VH le plus proche de *R* n'est pas judicieux. Illustration Figure 19 : le point *R* a été obtenu grâce à l'algorithme VHBS en utilisant les caméras C_1 et C_2. Relativement à celles-ci, *R* est situé « à l'avant » de l'objet à modéliser. Pour que la structure de VH se rapproche de celle de l'objet à modéliser, il faut donc déplacer un point situé « à l'avant » de VH. Or, le point le plus proche (*Q*) se situe « à l'arrière » de VH, il ne doit donc pas être choisi. Il faut préférer à *Q* le point le plus proche de *R* à la fois visible par C_1 et C_2 : c'est le point *P*.
On définit ainsi un *attracteur* : un couple de points : (*P*, *R*) où *R* représente la position réelle d'un point de l'objet à modéliser, et *P* le point du maillage qui devra occuper cette position. La calcul de *P* nécessite donc la connaissance des deux caméras qui ont permis de calculer *R*.

4.5 - Déformation du maillage

Pour modéliser l'objet 3D, nous disposons de deux informations différentes. Premièrement, le VH définit un nuage de points très dense, dont les positions sont correctes dans les zones convexes, et incorrectes dans les zones concaves. Si le maillage contient *N* points différents, notons cet ensemble : $\Omega=\{M^i, i=1..N\}$. Deuxièmement, l'algorithme VHBS fournit un ensemble d'attracteurs épars, qui sont tous corrects. Si *n* attracteurs différents ont été créés, notons-les : $\{(P^i,R^i), i=1..n\}$. Les éléments du second ensemble doivent modifier ceux du premier ensemble par une transformation 3D.
Les déformations à appliquer aux points $\{P^i, i=1..n\}\subset\Omega$ est connue : chacun de ces points subit la déformation : $F^i = R^i - P^i$. Les déformations à appliquer aux autres points de Ω doivent être déduites par interpolation.

Le maillage est déformé par RBF (Radial Basis Function), utilisées en raison de l'aspect 'lissé' des déformations effectuées [Turk 99]. Tout point *M* de l'espace subit une déformation obtenue par combinaison linéaire des distances euclidiennes (fonction radiale choisie) entre ce point *M* et les n points *Pi*. Le point *M* subit donc la transformation :

$$\begin{aligned} F(M) &= \sum_{i=1}^{n} \begin{bmatrix} A_x^i & A_y^i & A_z^i \end{bmatrix}^t d(M, P^i) \\ &= \sum_{i=1}^{n} A_x^i d(M, P^i)\vec{i} + \sum_{i=1}^{n} A_y^i d(M, P^i)\vec{j} + \sum_{i=1}^{n} A_z^i d(M, P^i)\vec{k} \\ &= F_x(M)\vec{i} + F_y(M)\vec{j} + F_z(M)\vec{k} \end{aligned} \qquad (2)$$

où *d* est un opérateur de calcul de distance (euclidienne ici) et où les *3 n* ' A_w^i ' sont des inconnues à déterminer. Or, chacun des *n* attracteur permet d'écrire les trois équations de déformation suivantes :

$$F_x(P^i)=R_x^i \; ; \qquad F_y(P^i)=R_y^i \; ; \qquad F_z(P^i)=R_z^i$$

On dispose donc de *3 n* équations et *3 n* inconnues, que l'on peut représenter sous la forme suivante :

$$\begin{bmatrix} d\left(P^1,P^1\right) & \dots & \dots & \dots & d\left(P^n,P^1\right) \\ d\left(P^2,P^1\right) & \dots & \dots & \dots & \dots \\ \dots & \dots & d\left(P^i,P^j\right) & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ d\left(P^n,P^1\right) & \dots & \dots & \dots & d\left(P^n,P^n\right) \end{bmatrix} \begin{bmatrix} A_x^1 & A_y^1 & Az \\ \dots & \dots & \dots \\ A_x^i & A_y^i & A_z^i \\ \dots & \dots & \dots \\ A_x^n & A_y^n & A_z^n \end{bmatrix} = \begin{bmatrix} R_x^1-P_x^1 & R_y^1-P_y^1 & R_z^1-P_z^1 \\ \dots & \dots & \dots \\ R_x^i-P_x^i & R_y^i-P_y^i & R_z^i-P_z^i \\ \dots & \dots & \dots \\ R_x^n-P_x^n & R_y^n-P_y^n & R_z^n-P_z^n \end{bmatrix}$$

Notons : **D.A=F** . La matrice inconnue **A** peut alors être déterminée (pivot de Gauss), nous obtenons alors les coefficients ' A_w^i 'de l'équation (2). Notons que si deux points P_i et P_j identiques sont utilisés dans le calcul de **D**, **D** n'est pas inversible et **A** ne peut être évaluée. Ces points sont donc détectés et éliminés du système.

5 - Résultats

Ces résultats sont produits sur des PC munis des logiciels Visual C++©, 3D studio max 4 © et du SDK associé. Les algorithmes sont développés en C++, le logiciel 3D studio max étant utilisé pour visualiser les données 3D et appliquer des fonctions 3D de haut niveau, comme l'intersection d'un maillage et d'un rayon optique.

5.1 - 'Visual Hull'

La *Figure 13* représente la scène 3D calibrée : on peut observer la position des caméras (en haut à gauche) et les images prises par 3 caméras différentes. Les 'Visual Hulls' (VH) doivent se superposer aux silhouettes du joueur s'ils sont bien construits.
C'est ce que l'on observe sur la *Figure 2,* avec un VH de résolution 12 cm, obtenu avec un octree de profondeur 4 et dont la longueur du côté du premier cube est de 205 cm. Le VH de la *Figure 4* est plus précis, puisque sa résolution est de 0.8 cm, pour une profondeur (maximale) de 9 et un cube initial de côté de longueur 205 cm.

Estimons visuellement la précision du VH de résolution maximale. Dans la position d'origine *(Figure 4*), le VH semble très précis. On peut observer que les zones concaves comme le cou, le dos, les fesses, l'arrière du genou sont parfaitement représentées. C'est normal car cette vue a été utilisée pour créer le VH : le découpage du volume est donc correct ici. Depuis un angle de vue d'une caméra de synthèse où les zones concaves sont visualisées de face et non de profil *(Figure 3*) le VH semble aussi très précis. Par contre, depuis un autre angle de vue créé par une seconde caméra de synthèse où les concavités sont vues de profil *(Figure 5*), on constate que les zones concaves sont mal modélisées. En effet, le bras du personnage à modéliser est plié, ce qui crée une concavité, et ce volume est occupé par le VH. C'est le problème du point B de la *Figure 1.*

On peut noter l'influence importante de la précision de la segmentation en observant la *Figure 4*, où une petite partie du pied du joueur est visible sur l'image en arrière-plan, alors qu'elle devrait être cachée par le VH. Le schéma de la *Figure 18* montre bien que si la silhouette est correcte dans une image, mais qu'elle est creusée dans une autre image, une partie du volume du VH sera manquante. En effet, pour créer un voxel, il faut que celui-ci se projète dans toutes les silhouettes, ce qui ne peut se produire en cas d'erreur de segmentation.

La résolution maximale théorique du VH calculée grâce à la formule présentée en 4.3.2 - est : δ_{theo} =0.5 cm. On constate que la valeur $\delta_{pratique}$ (longueur du côté du plus petit cube) mesurée dans la scène est de 0.8 cm. Les résultats sont donc cohérents.

5.2 - Visibilité

La visibilité est calculée à l'aide de la méthode présentée en 8.2.2 - . Pour cela, on affecte une couleur à chaque face du maillage. La *Figure 20* nous montre un maillage de 406.692 faces, auquel 406.692 couleurs ont été affectées. Dans la *Figure 21,* les faces coloriées sont celles qui sont visibles simultanément par les caméras numéro 1 et 6.
La résolution du VH est si élevée que certaines faces occupent à peine un pixel de l'image. Afin d'effectuer des calculs de visibilité très précis, nous avons travaillé sur des images de taille 5000 x 4074 plutôt que sur des images de taille 720 x 576, afin que toutes les faces soient correctement représentées.

5.3 - Détection des points caractéristiques.

Les points caractéristiques présentés sur la *Figure 16* sont détectés avec la méthode présentée en 4.4.1 - . Les valeurs des paramètres utilisés sont : disparité de 0.1, distance de 3 pixels minimum entre les points caractéristiques, distance de 2 pixels minimum par rapport au contour le plus proche.

5.4 - VHBS

Les zones concaves situées sur le joueur de golf que nous étudions sont visibles principalement sur la vue de face. Comme nous ne disposons pas de deux vues frontales, indispensables à la stéréo-vision, nous ne pouvons pas appliquer notre algorithme au joueur de golf. Aussi avons-nous modélisé le problème du bras plié. Sur la *Figure 22,* on aperçoit 6 caméras, un parallélépipède tordu vu de haut (en marron) qui a la forme d'un bras plié, et le VH de ce parallélépipède créé à partir des 6 caméras. L'algorithme de VHBS est testé sur ce modèle.

Nous essayons dans un premier temps de retrouver la position 3D d'un unique point caractéristique (*Figure 23*). Le point choisi dans l'image *Warped offset* (ou *mapped & render*) a pour coordonnées (438,461). Ce point est retrouvé dans l'image d'origine, à la position (353, 477), sans avoir fourni de disparité visuelle maximale, i.e. de manière totalement automatique. On constate donc que, malgré le caractère très approximatif du VH, et donc l'importante disparité (une centaine de pixels) les véritables coordonnées 3D ont été retrouvées. Ce résultat est obtenu avec les méthodes de calcul de correspondances stéréo classiques SAD (*sum of absolute difference*) et SSD (*sum of square difference*), rappelées en 8.3.2 -

Ce résultat étant très satisfaisant, nous appliquons l'algorithme à l'ensemble des points caractéristiques détectés sur une image représentant la partie concave de l'objet. En bas de la *Figure 24* on peut observer le VH, l'objet 3D à modéliser et les positions 3D estimées des points caractéristiques. En haut, pour vérifier la précision du positionnement de ces points, le VH n'est pas représenté. On constate que tous les points sont placés à la bonne profondeur sur l'objet 3D : leur position a donc été parfaitement retrouvée, et ce de manière totalement automatique.

On constate ici toute l'efficacité de la méthode, qui, partant d'une estimation très grossière de l'objet 3D (le VH) parvient à calculer avec une bonne précision les positions de points situés sur la surface de l'objet à modéliser (donc situés dans le VH).

5.5 - Déformation du maillage

Pour le moment, la création des attracteurs a un défaut : dans le cas du parallélépipède, quelques faces sont étiquetées comme « visibles » sur la partie supérieure du maillage (zone entourée de la *Figure 30*). Les sommets de ces faces rentrent donc en compte lors de la recherche du point visible le plus proche, ce qui empêche un point situé à l'avant du VH d'être choisi.
De ce fait, les déformations sont incorrectes : *Figure 25*

Afin de tester la fonction de déformation, nous avons déformé un cylindre à l'aide de points éloignés de celui-ci (*Figure 26* et *Figure 27*) : la fonction de déformation est correcte. Il nous faut donc améliorer la création des attracteurs.

6 - Conclusion

L'objectif de ce stage était de proposer une méthode de création de l'enveloppe corporelle d'un golfeur à partir d'un jeu de photos et basée sur approche volumique.

L'analyse de nos contraintes (précision et automatisation), d'une part, et des méthodes utilisées par la communauté scientifique, de leurs défauts et de leurs avantages, d'autre part, nous a permis de mettre au point une méthode de création de maillage innovante, en utilisant deux informations fournies par les photos du joueur : ses silhouettes dans les diverses vues, et sa texture.

La méthode proposée permet de modéliser des objets génériques connexes, bien qu'elle ait été conçue dans l'esprit de modéliser un être humain. De plus, dans un soucis d'automatisation, les paramètres utilisés sont fixés une fois pour toutes, hormis l'éloignement minimum des points caractéristiques et la taille des fenêtres de calcul des correspondances stéréo, qui dépendent de la densité des informations fournies par les photos.

Ce rapport apporte une contribution à deux domaines de la vision. Le premier domaine est la stéréo-vision : une solution est proposée au problème des appariements dans le cas de disparités trop importantes (deux points de vue formant un angle de plus de 45 degrés par exemple). Pour utiliser la méthode VHBS, il suffit de disposer autour du sujet des caméras dont on connaît les paramètres. Le second domaine est la reconstruction 3D à partir de photos : rares sont les méthodes précises ne faisant pas de suppositions à priori sur l'objet à modéliser[16]

Certains travaux restent à effectuer pour parfaire cette technique.

On peut supprimer l'aliasing dû à la présence de voxels en utilisant l'algorithme éprouvé du 'Marching Cube' [Lorensen 87]. L'algorithme très rapide utilisé ici était temporaire, et permettait d'accorder plus de temps aux problèmes fondamentaux.

L'amélioration des attracteurs est, elle, primordiale, car ceux-ci déterminent l'efficacité de la méthode. En effet, la méthode VHBS permet de trouver des positions 3D précises mais ne donne pas encore le vecteur de déformation correct : quel point P^i du maillage doit être attiré ? Nous avons plusieurs pistes pour résoudre ce problème : l'une d'elles consiste à définir une mesure de *densité de visibilité des faces.* On remarque en effet que les points situés dans des zones visibles denses sont généralement plus efficaces que ceux situés dans les zone de faible densité de visibilité (*Figure 30*).

Enfin, il est possible d'obtenir des résultats encore plus précis en utilisant un modèle dont chaque face contient la méta-donnée booléenne suivante : « cette zone est-elle concave ou convexe ? ». Dans notre cas, un modèle humain peut être utilisé après avoir obtenu le VH, pour supprimer une partie du volume des zones concaves : en plaçant des ellipsoïdes dans le VH, et sachant qu'elles sont en contact avec les zones convexes, grâce aux propriétés de symétrie du corps humain, on peut déduire la part de volume à supprimer. Cela réduirait davantage les disparités dans le cas du VHBS, et augmenterait encore la précision de la modélisation (segment épipolaire plus court).

Une fois que la création des attracteurs sera améliorée, une version de ce rapport condensée et traduite en anglais sera proposée au comité CVPR 2004 [17], l'article devant parvenir avant le 19 novembre. L'article s'inscrira dans le sous-thème : [Image-based modeling].

[16] La segmentation est *manuelle*, mais une capture de donnée avec modifications colorimétriques aurait permis d'effectuer une bonne segmentation automatique.

[17] http://cvl.umiacs.umd.edu/conferences/cvpr2004

7 - Références bibliographiques

[Debevec 96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and Rendering Architecture from Photographs. In SIGGRAPH '96, August 1996.

[Cheung 00] Cheung, Kanade, Bouguet, Holler, *A real time system for robust 3D voxel reconstruction of human motions,* Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition, #12, pp. 714 – 720, 2000

[Faugeras 93] O. D. Faugeras. Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press, Cambridge, Mass., 1993

[Hilton 00] Hilton, Beresford, Gentils, Smith, Sun, Illingworth, *Wole-body modelling of peaple from multiview images to populate virtual worlds,* Springer Eds, 2000

[Lewis 00] Matthew Lewis, *Evolving Human Figure Geometry*, Technical report, May 2000.

[Li 02] Li, Schirmacher, Magnor, Seidel. *Combining Stereo and Visual Hull Information for On-line Reconstruction and Rendering of Dynamic Scenes*. IEEE 2002 Workshop on Multimedia and Signal Processing., 2002

[Lorensen 87] W. Lorensen and H. Cline, Marching cubes: A high resolution 3-D surface construction algorithm, Computer Graphics, vol. 21, 1987, 163-169

[Moezzi 00] Moezzi S., Tai L.-C., Gerard P. Virtual View Generation for 3D Digital Video. IEEE Multimedia, 4(1):18-26, January 1997.

[Slabaugh 01] Slabaugh, Culbertson, Malzbender, Schafer, *A Survey of Methods for Volumetric Scene Reconstruction from Photographs*, VG, pp. 81-100, 2001

[Szeliski 93] Szeliski, *Radip octree construction from image sequence*, CVGIP : image understanding, 58(1), pp 23-32, 1993

[Turk 99] G. Turk and J. O'Brien, *Shape transformation using variational implicit functions*, SIGGRAPH' 99, 335-342.

[Sun 99] Sun, Hilton, Smith, Illingworth, *Layered animation of captured data*, animation and simulation '99, pp 145-154, 1999

[Weik 00] Weik, Wingbermühle, Niem, *Creation of flexible antropomorphic models for 3D videoconferencing using shape from silhouettes*, The journal of visualization and computer animation, #11, pp 145-154, 2000

8 - Annexe

8.1 - Illustrations

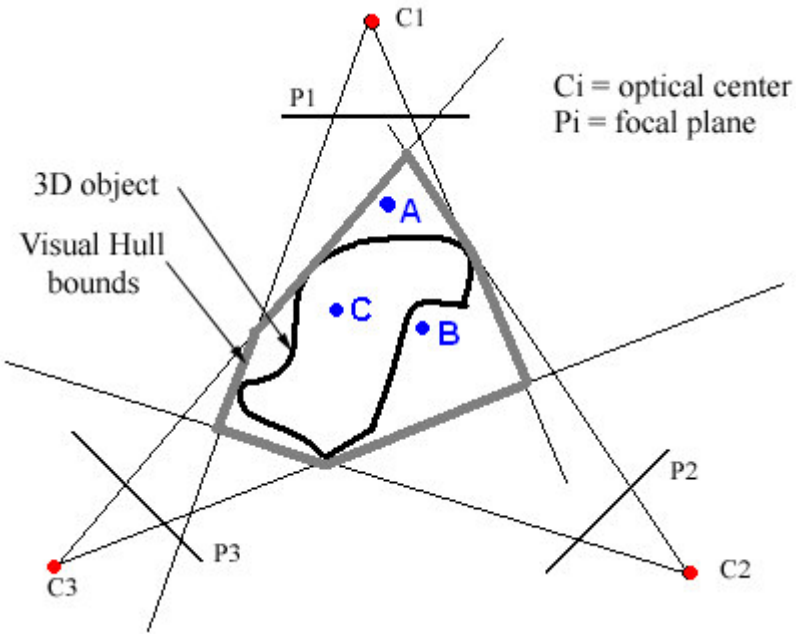


Figure 1 – Principe du Visual Hull

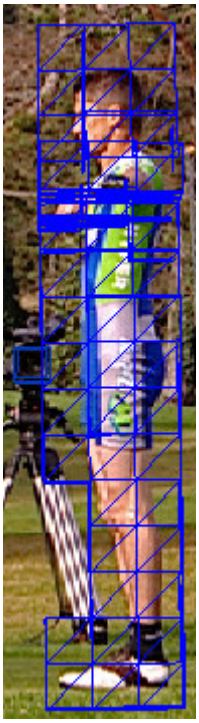


Figure 2 – 'Visual Hull' d'un golfeur, faible résolution. Point de vue d'une caméra réelle.

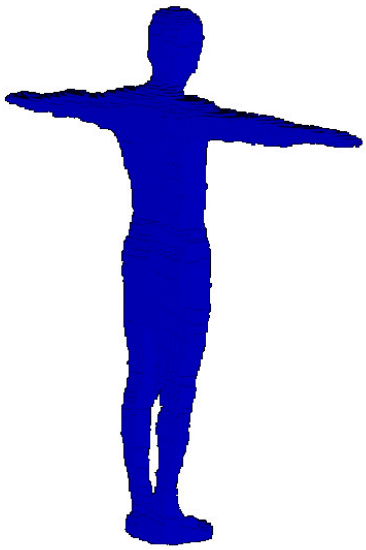


Figure 3 – 'Visual Hull' de résolution maximale – Point de vue d'une caméra de synthèse.

Figure 4 – 'Visual Hull' de résolution maximale placé dans la scène

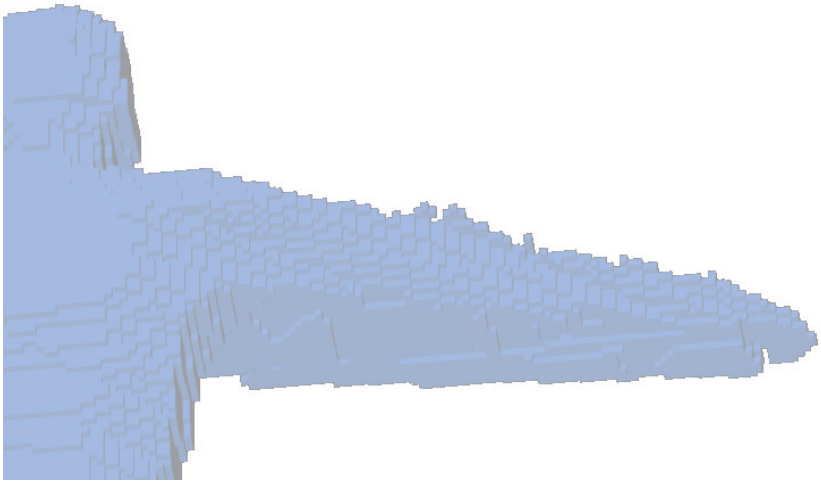


Figure 5 – Erreurs de modélisation des parties concaves avec un 'Visual Hull'.

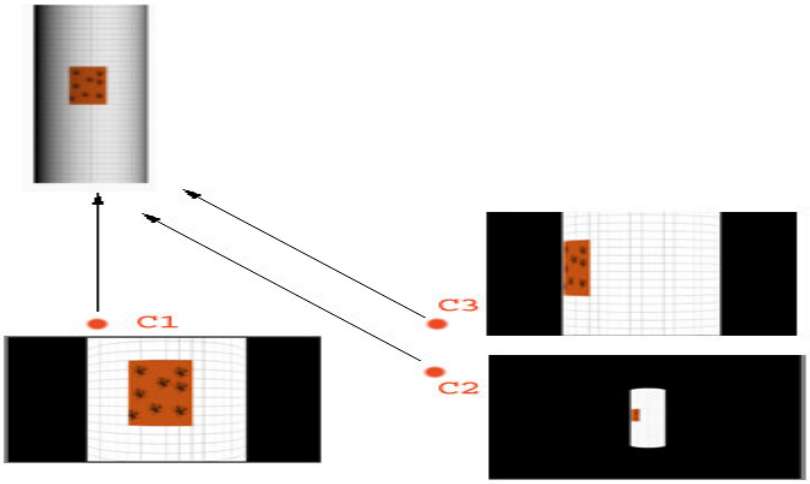


Figure 6 – Difficultés dans le calcul des correspondances – Motif observé par 3 caméras.

(a) *Key Image* (b) *Warped Offset Image* (c) *Offset Image*

Figure 7 – MBS, images : vue 1 (a), vue 2 (c), et warping de la vue 2 dans la vue 1 (c). [Debevec 96]

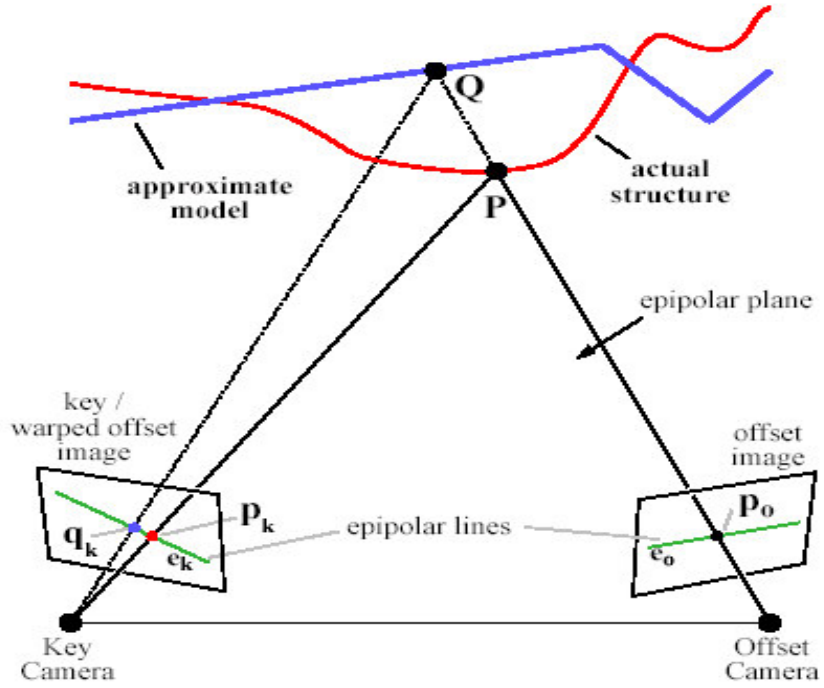


Figure 15: *Epipolar geometry for model-based stereo.*

Figure 8 – Principe MBS [Debevec 96]

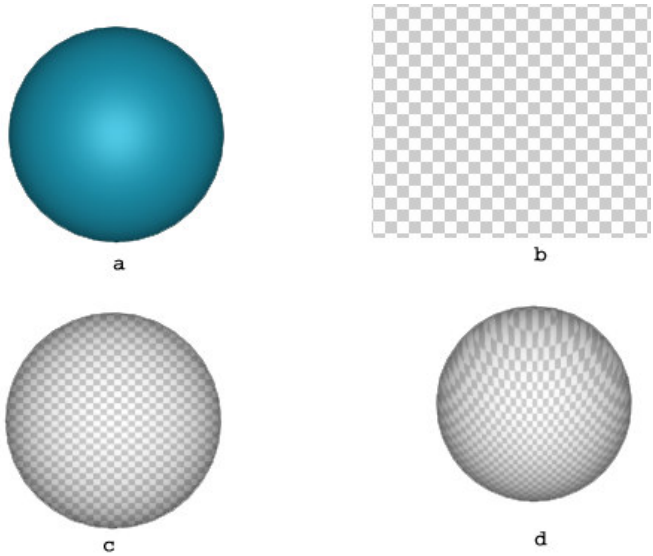


Figure 9 - Camera mapping : sur a, avec motif b, et camera vue c. d = nouveau point de vue

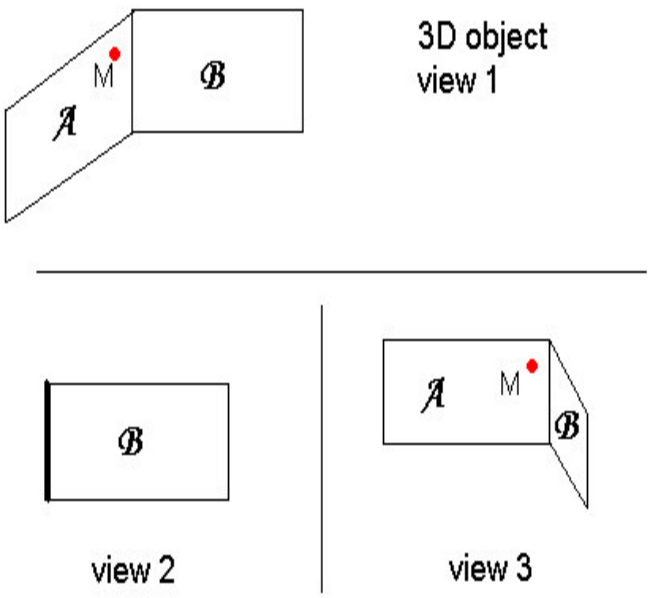


Figure 10 – Problèmes de visibilité du point M

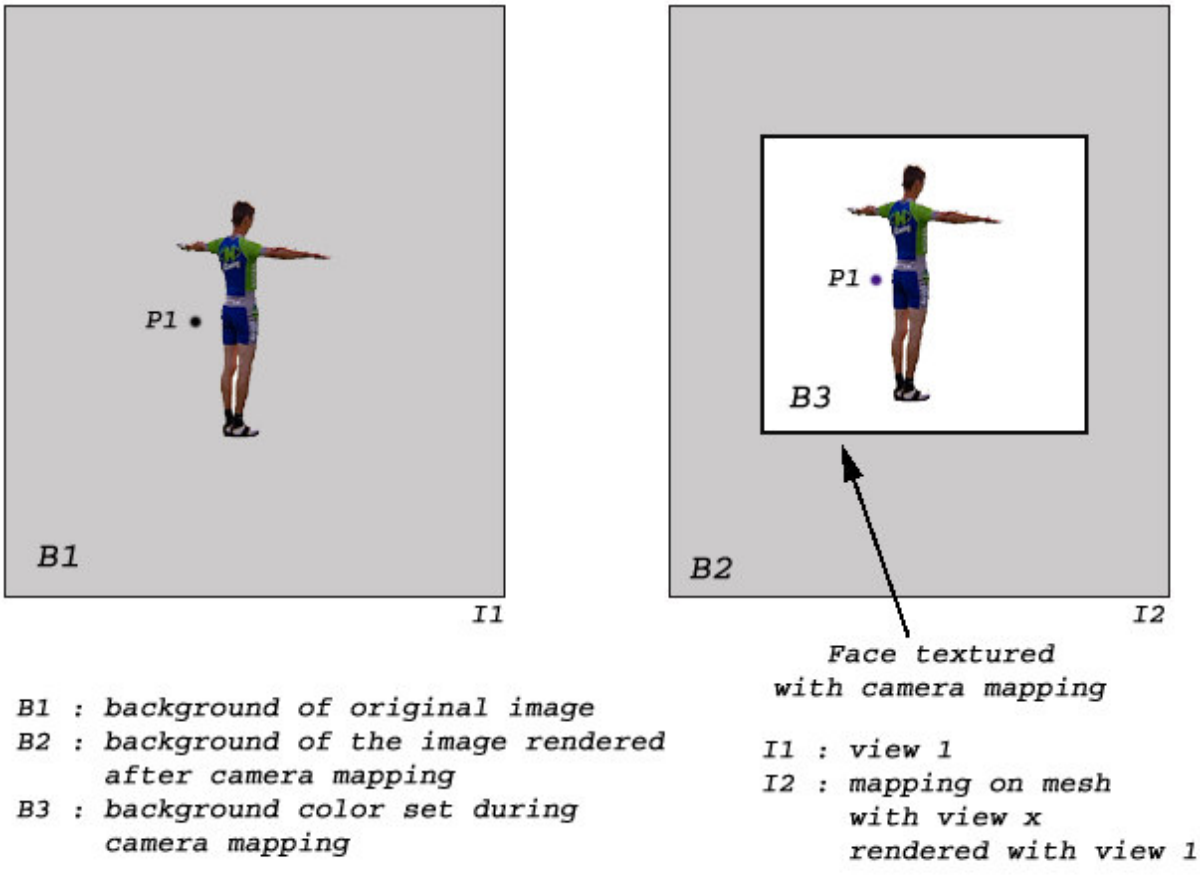


Figure 11 – Problème de visibilité du point P1

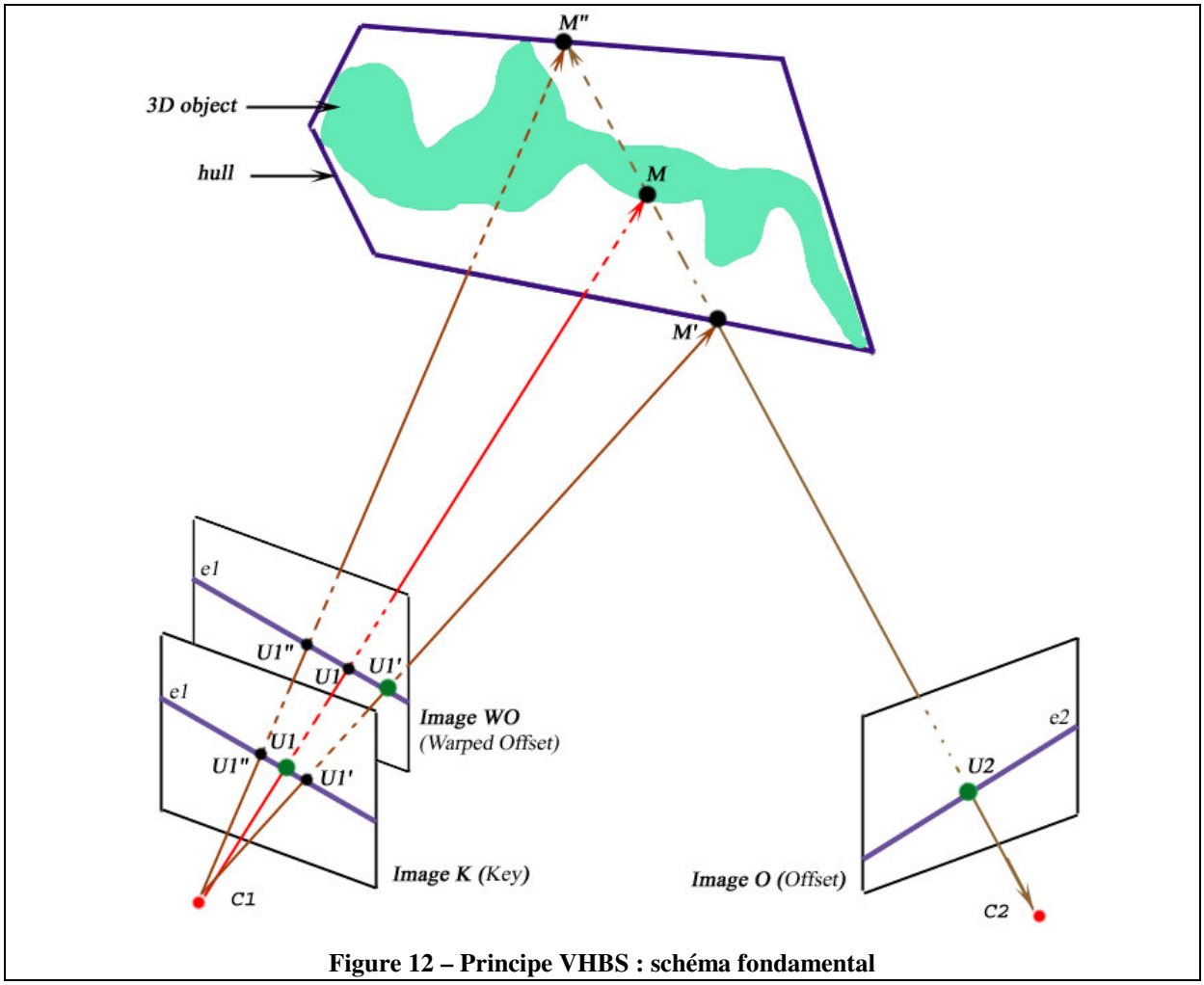


Figure 12 – Principe VHBS : schéma fondamental

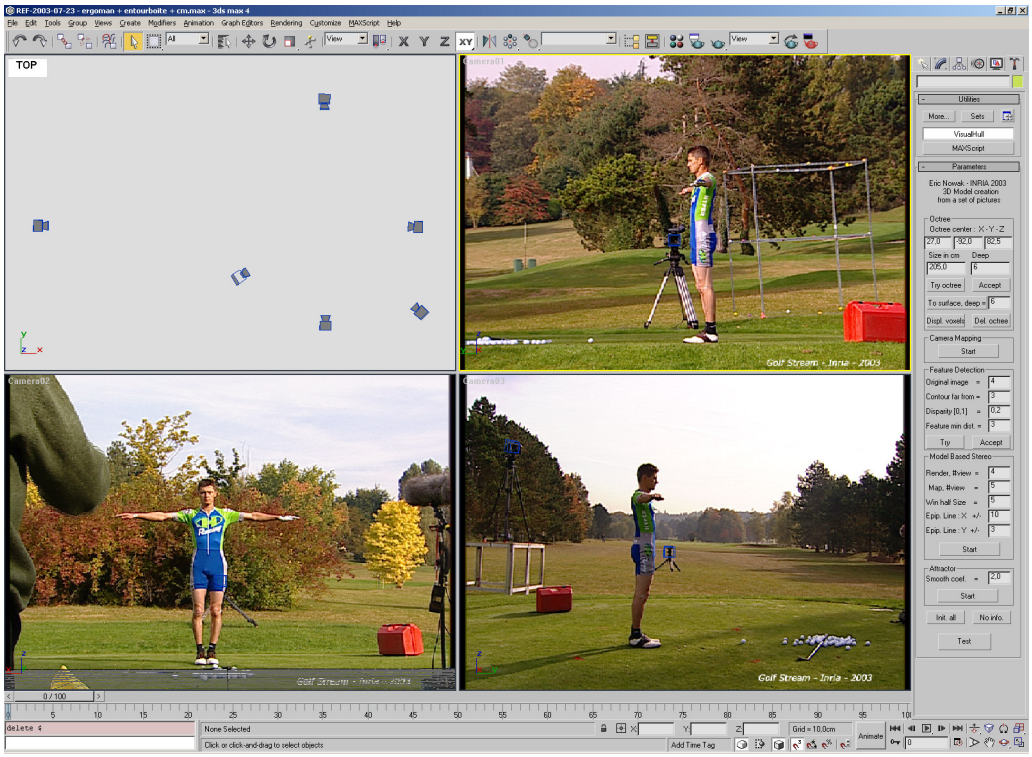


Figure 13 – Scène calibrée – 3D studio max – Position des caméras (en haut à gauche) et 3 vues.

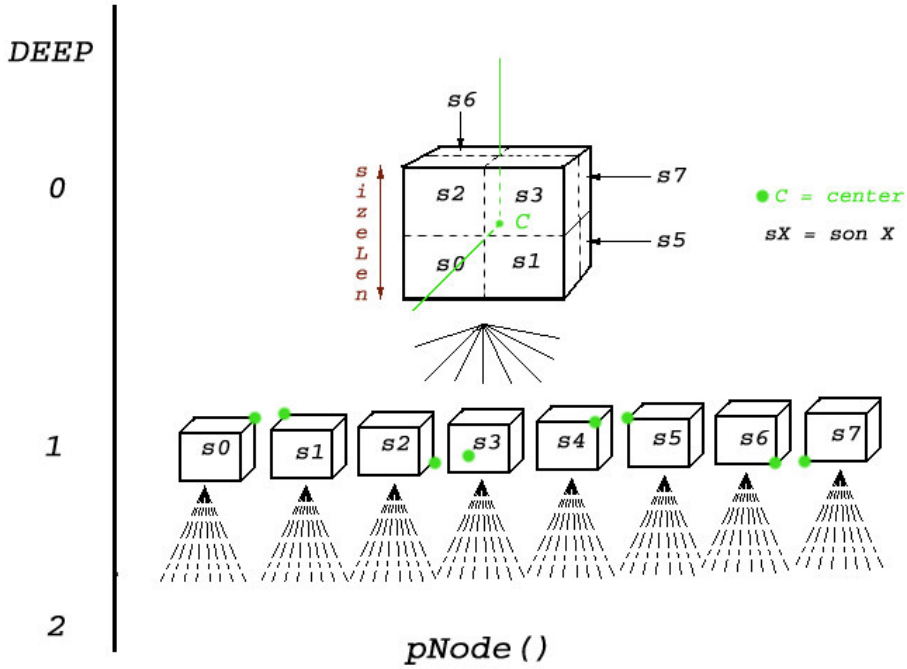


Figure 14 – Principe de l'octree

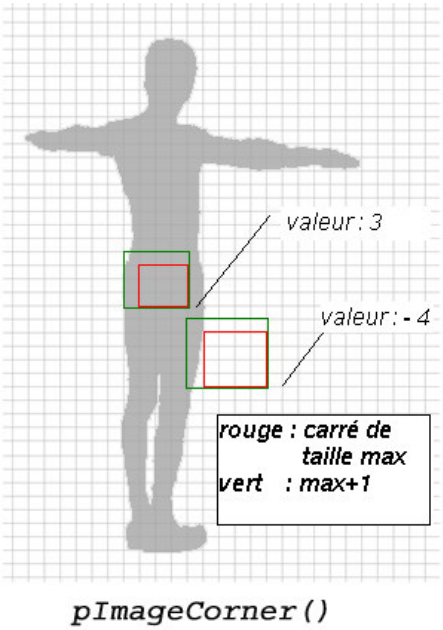


Figure 15 – Carré de taille maximale entièrement à l'intérieur / extérieur de la silhouette

Figure 16 –Détection de points caractéristiques sur image Warped Offset

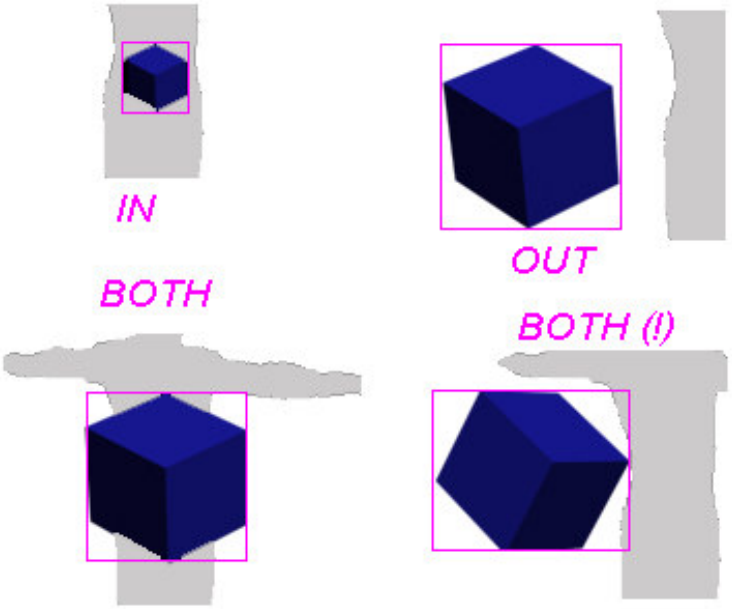


Figure 17 – Rectangle englobant de la projection d'un voxel ; positionnement par rapport à la silhouette

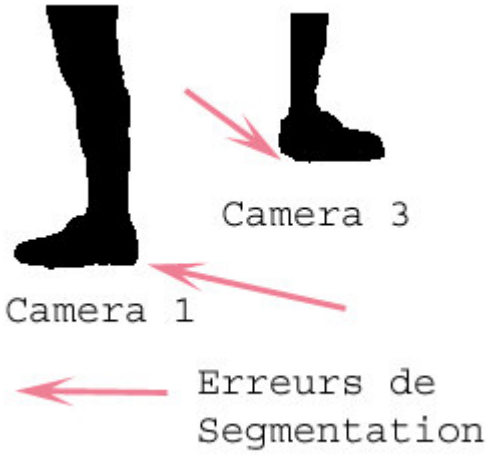


Figure 18 – Influence de la segmentation dans la construction du 'Visual Hull'.

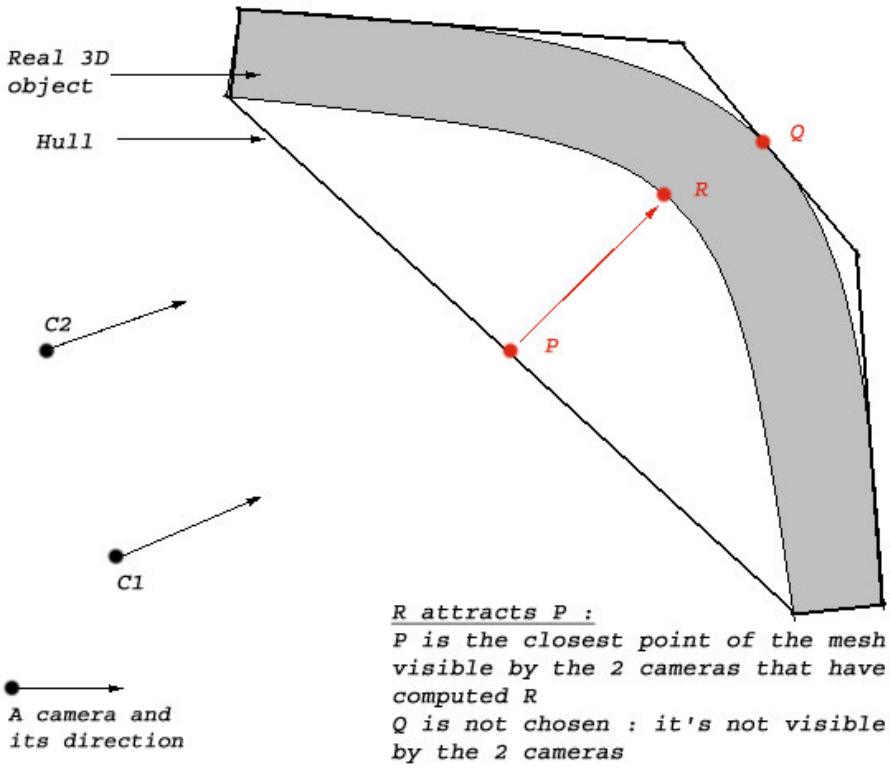


Figure 19 – Attracteur : choix du point du maillage à attirer

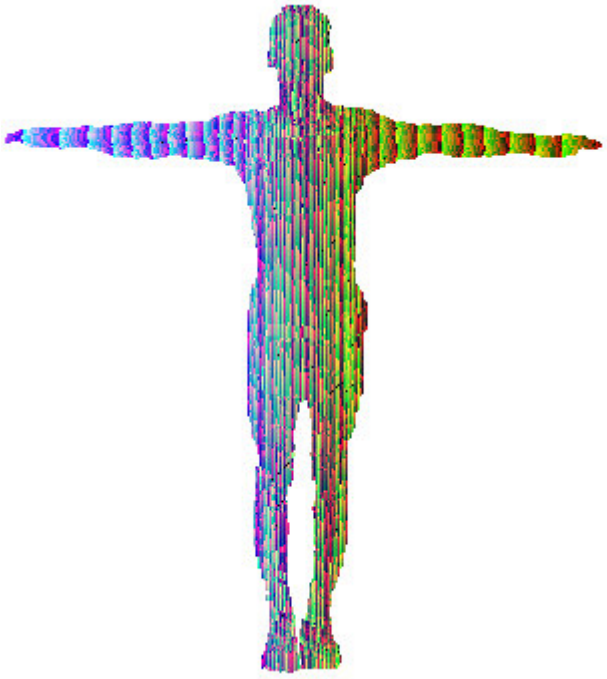


Figure 20 – Test de visibilité : une couleur par face (406.692 faces)

Figure 21 – Faces visibles dans la vue 1, toujours visibles dans la vue 6 (en couleurs)

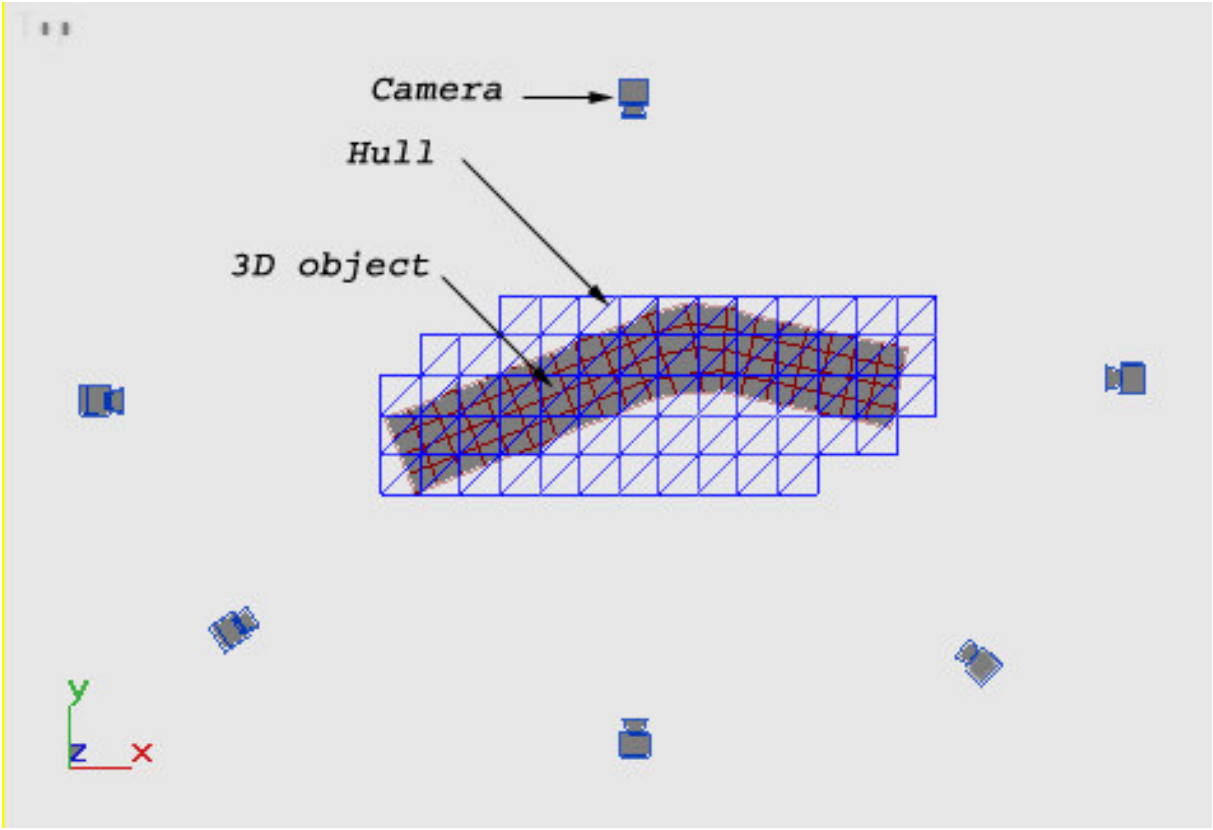


Figure 22 – Modélisation du problème : « zones concaves »

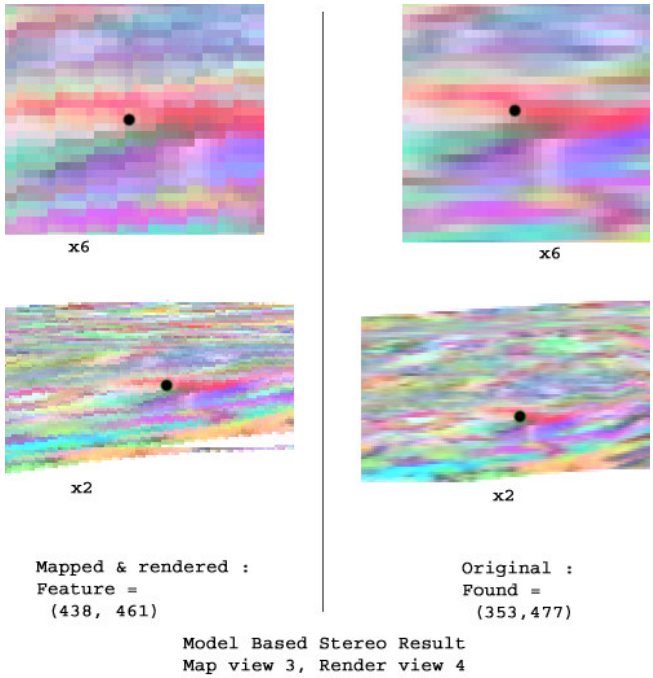


Figure 23 – VHBS : exemple pour un point caractéristique (noter : précision malgré la disparité)

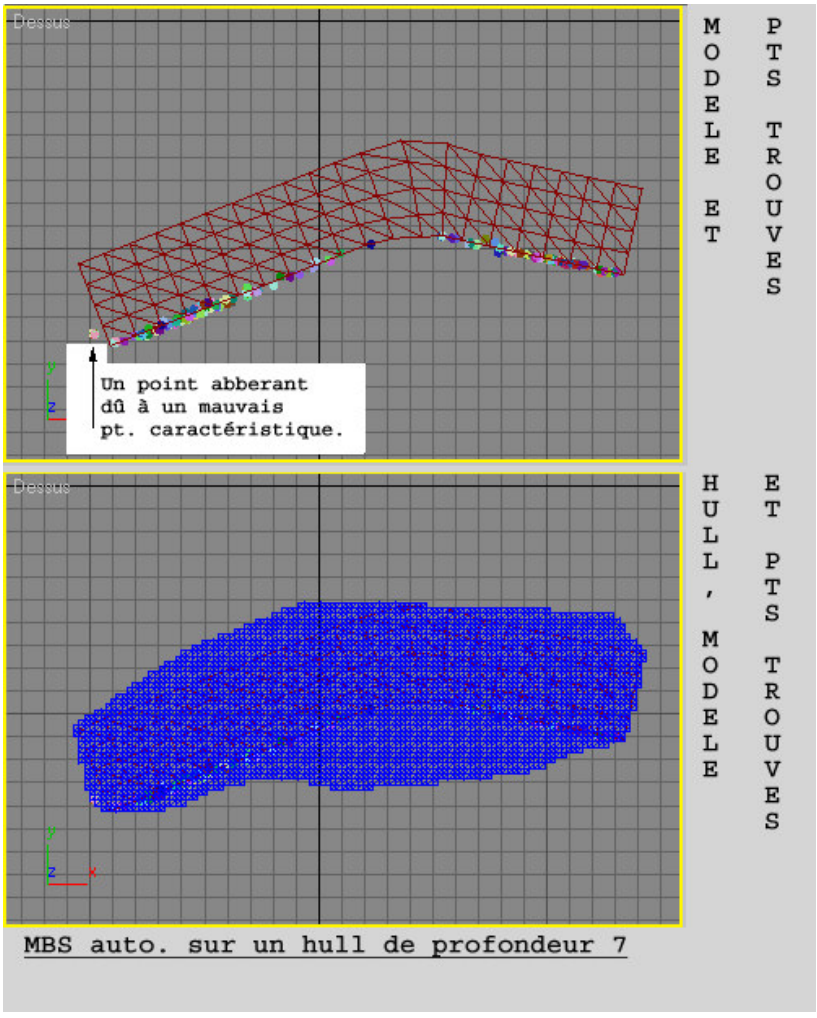


Figure 24 – VHBS sur plusieurs points caractéristiques, avec modélisation de zone concaves. Les positions 3D sont retrouvées avec une très bonne précision

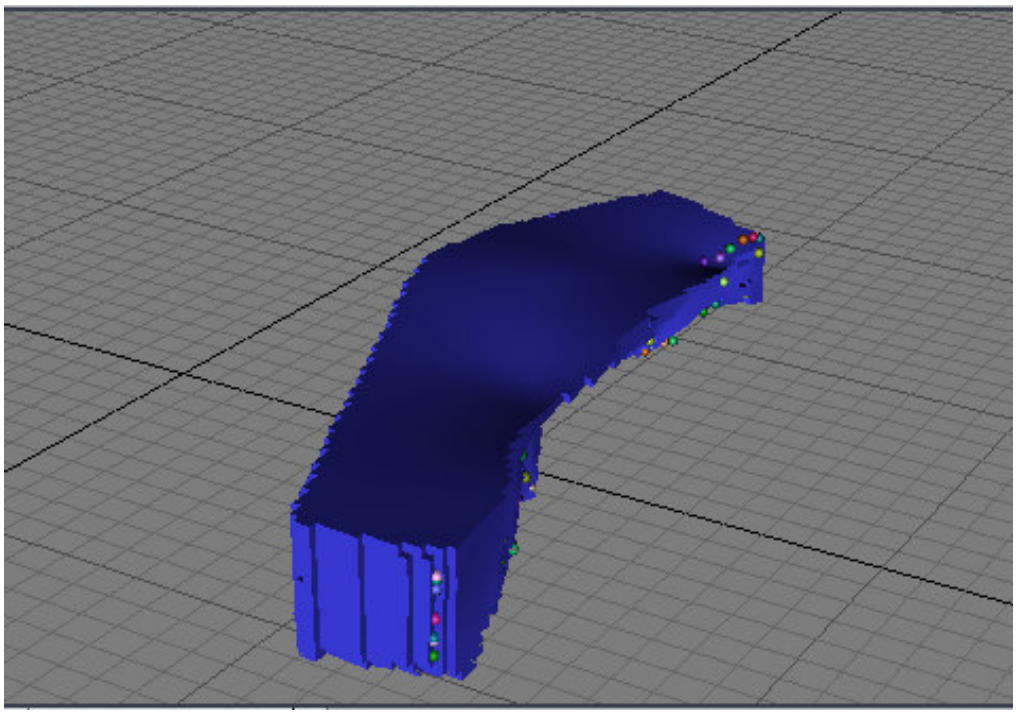


Figure 25 – Déformations avec RBF avec les données (‘Visual Hull’, attracteurs) de la Figure 24

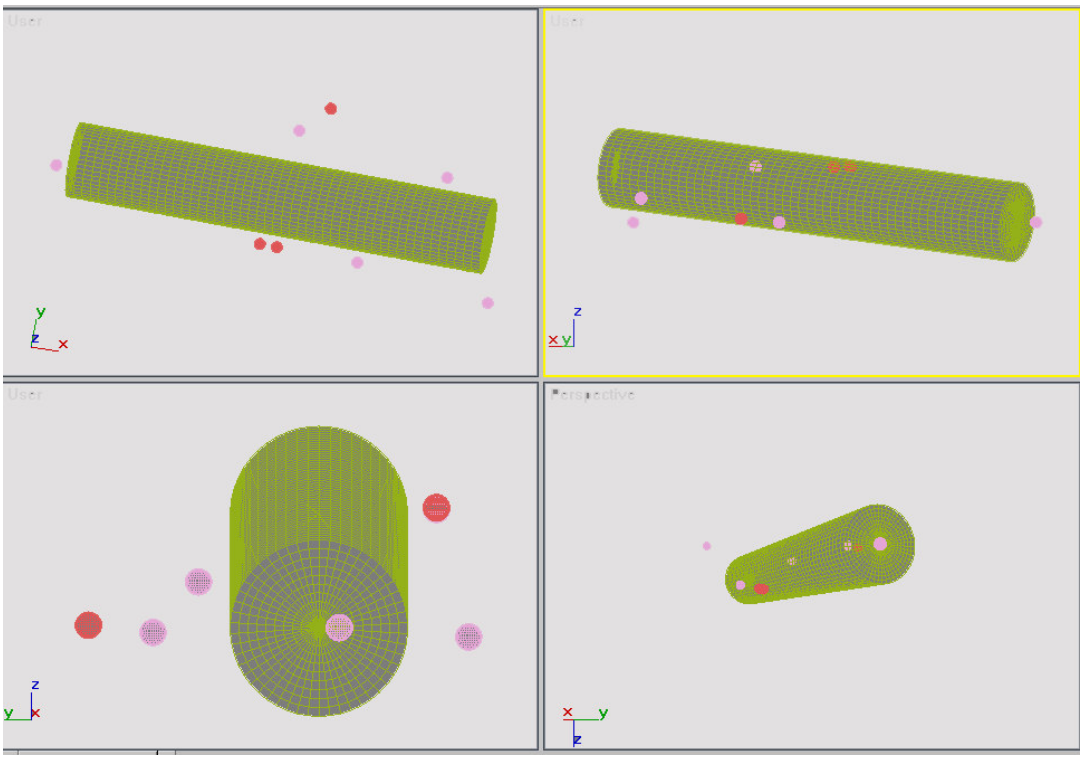


Figure 26 – Déformation d’un cylindre par des attracteurs

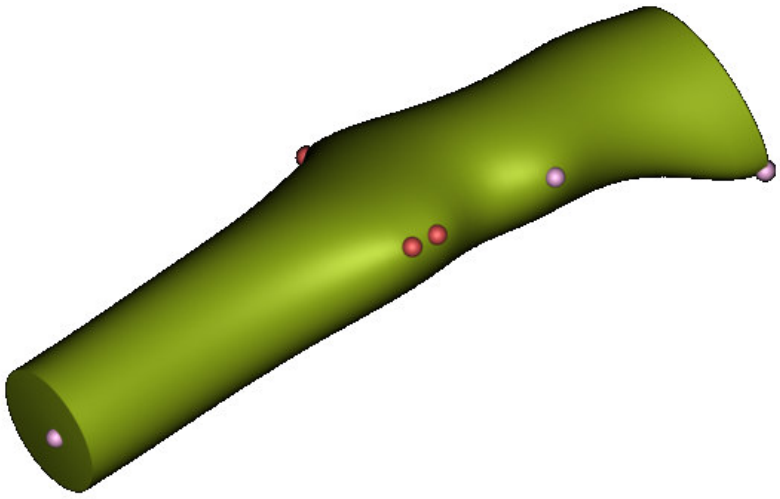


Figure 27 – Résultat de la déformation Figure 26

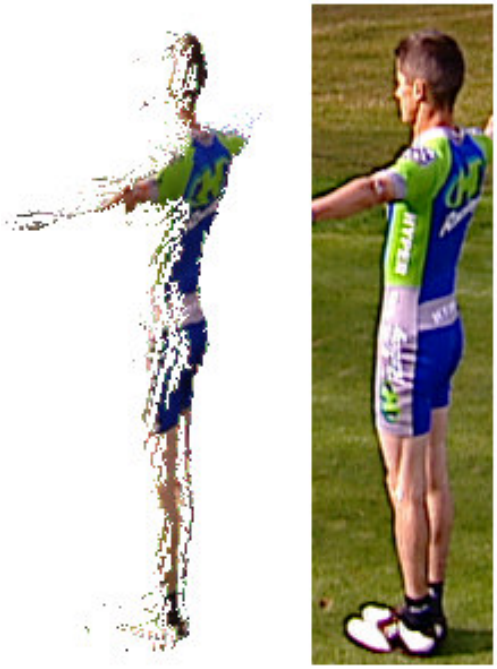


Figure 28 - Observation des disparités entre les parties visibles des images Key (à droite) et Warped offset (à gauche)

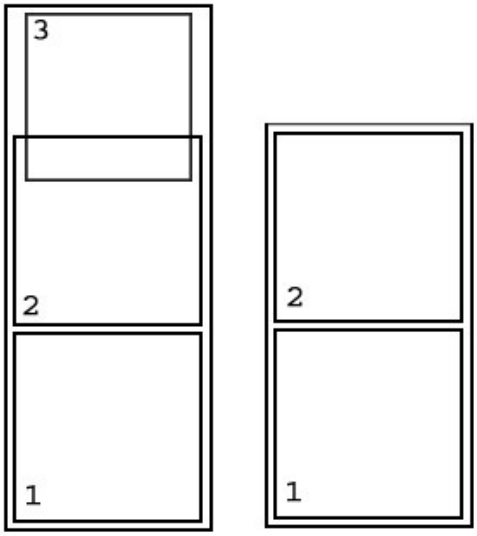


Figure 29 - Parcours d'un rectangle par des carrés (le dernier peut être superposé au précédent)

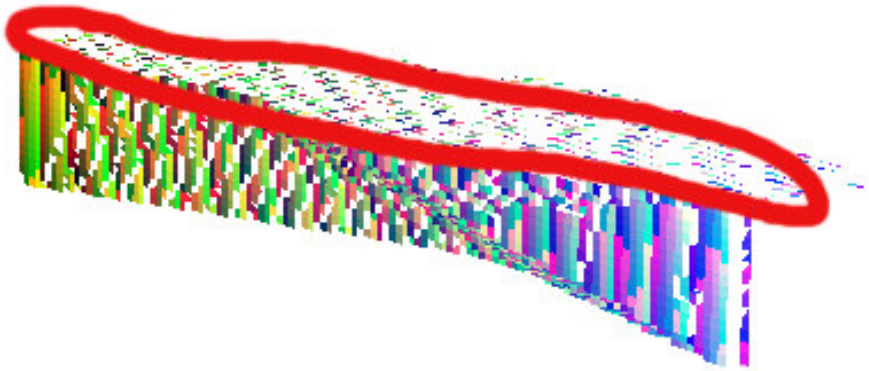


Figure 30 - Création des attracteurs : problème des faces visibles en haut du 'Visual Hull'

8.2 - Méthodes, démonstrations

8.2.1 - Méthode d'intégration du squelette dans l'enveloppe corporelle et de skinning automatique

On dispose de deux jeux de photos prises simultanément : le premier dans une posture, le second dans une autre posture. On dispose aussi du squelette 3D de la personne à modéliser (mesures, outils, etc.).

Le but de l'opération est de positionner le squelette correctement dans le maillage, c'est à dire retrouver les DOF (degrés de liberté).

Méthode proposée :

Notons S_{ij} l'ensemble des pixels occupés par la silhouette i dans la posture j.
Notons P_{ij} l'ensemble des pixels où se projète le 'Visual Hull' dans l'image i, pour la posture j.

1) Obtention d'une posture à peu près correcte (sans skinning):
Donner une valeur aléatoire aux DOF, ou une valeur approximative (ce deuxième choix nécessite un humain qui pré-positionne de manière très approximative le squelette).
Définir le déplacement des points du maillage en fonction du déplacement de l'os le plus proche.
Rechercher par recuit simulé les DOF permettant de passer d'une posture à l'autre avec le moins d'erreurs possible.
La fonction d'erreur est la taille des surfaces erronées, c'est une erreur de position :

$$err_{totale} = err_{pos} = \#\{P, \exists i, j, P,\ P \in (S_{ij} \cup P_{ij}), P \notin (S_{ij} \cap P_{ij})\}$$

2) Raffinement de la posture (avec skinning)
Maintenant que la posture est à peu près correcte, il faut faire intervenir le skinning.
Pour cela, il faut définir le déplacement des points du maillage en fonction du déplacement relatif d'un ensemble d'os. Cela permet d'éviter les cassures quand le coude se plie par exemple.
Les nouveaux coefficients à régler sont donc les poids de skinning : influence relative des os sur le maillage.
On définit une nouvelle fonction d'erreur :

$$err_{totale} = err_{pos} + A_{skin}$$

où A_{skin} est un terme à définir, dont le but est de minimiser les modifications dues au skinning.
Cela force l'algorithme à préférer une bonne position générale avec peu de skinning plutôt qu'une mauvaise position générale très skinnée.

Le terme A_{skin} peut être du type : $A_{skin} = \sum (1 - \omega_{ij})^2$ où les ω_{ij} sont les poids de skinning normalisés des sommets du maillage.

8.2.2 - Méthode de calcul de la visibilité des faces du maillage

On définit une face du maillage comme visible depuis une camera C si au moins *min* pixels sont visibles dans le rendu image effectué avec les paramètres de cette caméra.

On affecte donc une couleur à chaque face du maillage, on effectue un rendu, on compte le nombre de pixels d'une couleur donnée, ce qui permet de déterminer si la face correspondante est visible ou non.

Pour cela, il faut définir une bijection entre l'espace des couleurs et le numéro (l'index) de la face.

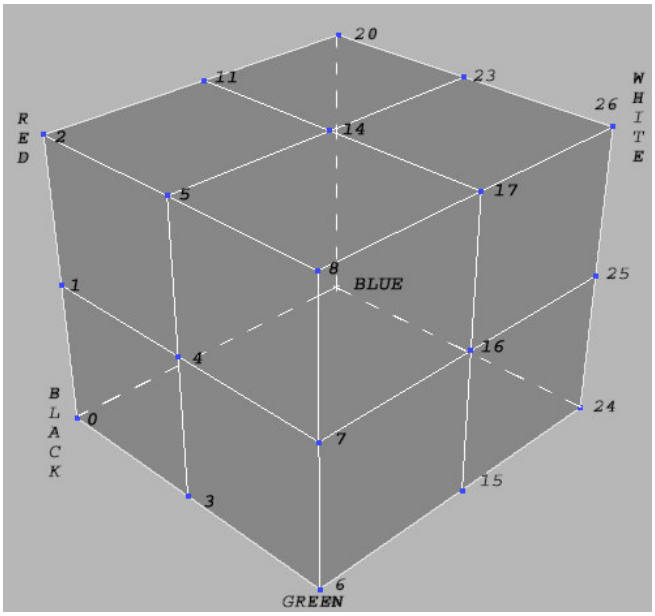
Pour cela, nous effectuons une subdivision régulière de l'espace RGB.

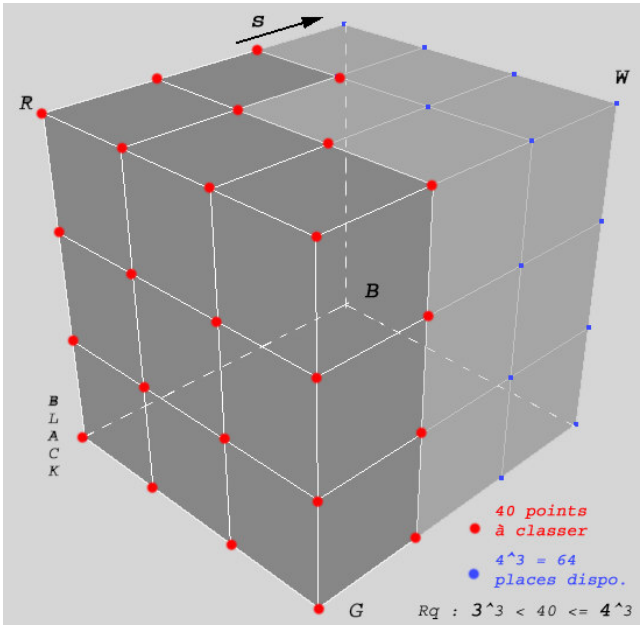
Pour répartir n points régulièrement dans un hypercube de côté 1 plongé dans un espace de dimension d, il faut subdiviser chaque côté de l'hypercube en : $n^{1/d} - 1$.

Par exemple, en dimension 3, on place 8 points en divisant chaque côté du cube en $8^{1/3} - 1 = 1$. Chaque côté du cube est donc conservé tel quel, et on obtient les 8 sommets.
Dans notre cas, nous voulons répartir n+1 points dans un espace de dimension 3, si n est le nombre de faces du maillage. En effet, le point de coordonnées (1,1,1) correspondant à la couleur 'Blanc' ne doit pas être utilisé, car il s'agit d'une couleur d'arrière plan pour le calcul de visibilité.

La figure de gauche ci-dessous nous montre comment sont réparties les couleurs d'un 'Visual Hull' (VH) de 26 faces. La figure de droite nous montre comment sont réparties les couleurs d'un VH de 40 faces. Tout une section du cube est vacante, cela signifie que l'espace n'est pas occupé au maximum. Nous effectuons une homothétie (de rapport 3/2) afin que la position (R,G,B) = (1,0,1) soit occupée.

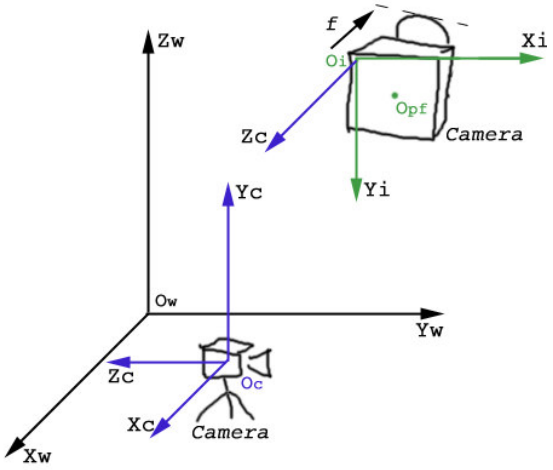
La fonction de correspondance entre l'ensemble des sommets définis et celui des couleurs associées est une bijection.





8.2.3 - Matrices de passage : W2P et P2W

Les notations utilisées sont celles de la partie 4.1 - et du schéma suivant :



Repère World : (O_w , x_w, y_w, z_w)
Repère Camera : (O_c , x_c, y_c, z_c)
Repère Image : (O_I , x_I, y_I)

W2P (world to pixels)

Soit $M = \mathbf{X_w}$ un point 3D non situé dans le plan focal de la caméra, dont les coordonnées sont exprimées dans le repère World en centimètres. Les coordonnées $\mathbf{X_{cam}}$ de M en centimètres dans le repère caméra sont :

$$\mathbf{X_c} = \mathbf{R}\left(\mathbf{X_w} - \mathbf{T_c}\right) = \begin{bmatrix} X_{c1} & X_{c2} & X_{c3} \end{bmatrix}^t$$

Comme l'axe z_c est dirigé vers l'arrière de la caméra, les coordonnées projectives du projeté de M dans le repère (O_{pf}, x_c, y_c) sont :

$$\begin{bmatrix} -f_c.X_{c1} & -pixR.f_c.X_{c2} & X_{c3} \end{bmatrix}^t$$

équivalentes à

$$\mathbf{X_{pf}} = \begin{bmatrix} \frac{-f_c.X_{c1}}{X_{c3}} & \frac{-pixR.f_c.X_{c2}}{X_{c3}} & 1 \end{bmatrix}^t$$

car X_{C3} est non nul, le point M n'étant pas situé dans le plan focal de la caméra. Ces coordonnées sont toujours exprimées en centimètres. Considérons alors le point 2D, dont les coordonnées, en centimètres, sont exprimées dans le repère (O_{pf}, x_c, y_c)

$$\mathbf{U_{cm}} = \begin{bmatrix} \frac{-f_c.X_{c1}}{X_{c3}} & \frac{-pixR.f_c.X_{c2}}{X_{c3}} \end{bmatrix}^t$$

Afin de convertir les coordonnées en pixels, il faut multiplier $\mathbf{U_{cm}}$ par le coefficient : $\frac{1}{c_{pc}} = \frac{L_p}{L_c}$.

On obtient donc le point :

$$\mathbf{U_{pix}} = \begin{bmatrix} \frac{-f_c.X_{c1}}{X_{c3}.c_{pc}} & \frac{-pixR.f_c.X_{c2}}{X_{c3}.c_{pc}} \end{bmatrix}^t = \begin{bmatrix} U_{pix\_x} & U_{pix\_y} \end{bmatrix}^t$$

Pour obtenir les coordonnées finales de ce point dans le repère image, il faut inverser l'axe des y et décaler l'origine du repère en haut à gauche de l'image. On obtient donc le point :

$$\mathbf{U} = \begin{bmatrix} U_{pix\_x} & -U_{pix\_y} \end{bmatrix}^t + \begin{bmatrix} \frac{L_p}{2} & \frac{H_p}{2} \end{bmatrix}^t$$

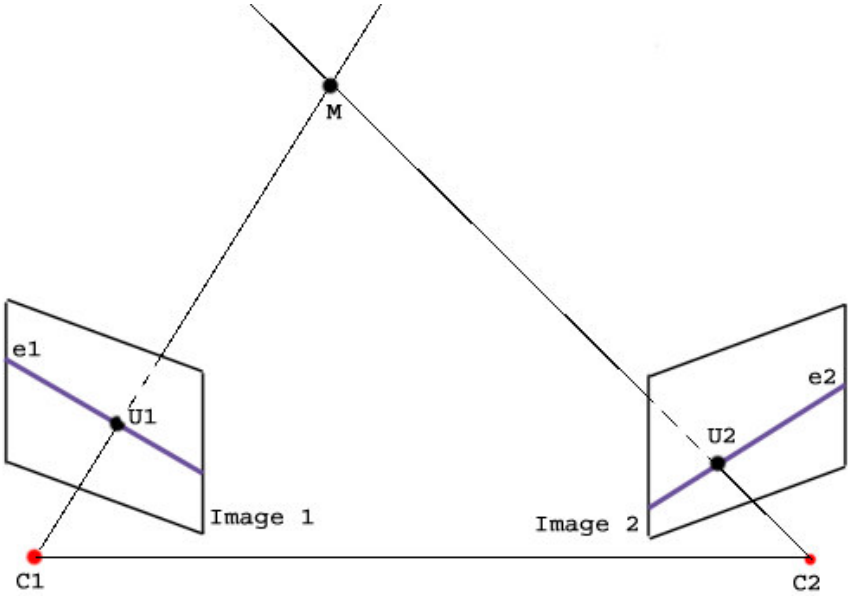
Le développement de la première formule fournie en 4.1 - donne le même résultat que les calculs précédents.

P2W (pixels to world)

La seconde formule donnée en 4.1 - se démontre en « remontant » les calculs ci-dessus.

8.3 - Rappels sur la stéréo-vision

8.3.1 - Géométrie épipolaire



Soient deux caméras *C1* et *C2*, photographiant une scène et produisant respectivement les images *Image 1* et *Image 2*. Le point 3D *M* se projète dans chacune de ces images, en *U1* et *U2* de dimension 2. La géométrie épipolaire précise la relation entre les points *U1* et *U2*.

Connaissant la position de *U1*, on cherche à obtenir des précisions sur la position de *U2*.
Le point M se situe sur le rayon optique (*C1, U1*), donc le projeté de *M* dans *Image 2* (i.e., *U2*) se situe sur le projeté de (*C1, U1*) dans *Image 2,* elle passe donc par les projetés de *C1* et de *U1* dans *Image2*. Cette droite se nomme la droite épipolaire, il s'agit ici de e_2.

8.3.2 - Calcul des correspondances

Considérons le schéma ci-dessus. Etant donné un point *U1*, la géométrie épipolaire nous affirme que le point correspondant *U2* se situe sur la droite épipolaire e_2. Comment trouver le point de cette droite correspondant à la projection de *M* ?

Comme *U1* et *U2* sont des projections du point *M*, on peut estimer que le voisinage de U1 et le voisinage de U2 sont similaires. Notons $\Delta(P1, P2)$ une fonction évaluant les différences entre deux fenêtres de même tailles, l'une centrée en *P1* élément de *Image 1* et l'autre centrée en *P2* élément de *Image 2*. Les voisinages des points *P1* et *P2* sont d'autant plus similaires que la valeur $\Delta(P1, P2)$ est faible. Le point *U2* peut donc être estimé comme :

$$U2 = \text{argmin} \{ \Delta(U1, U) \mid U \in e_2 \} \qquad (3)$$

Si différents voisinages de pixels situés sur e_2 sont identiques, l'équation (3) peut retourner un point U2 incorrect. C'est ce que l'on nomme un faux appariement.

La fonction Δ peut être définie comme la somme des valeurs absolues des différences des niveaux de gris des pixels des fenêtres centrées autour de *P1* et *P2* (SAD, sum of absolute difference) ou comme la somme du carré de ces différences (SSD, sum of square difference).

8.4 - Bibliographie sur le tracking, notion de 'Visual Hull'