

Silhouettes pour le Suivi Temps-Réel basé Modèle

Real-Time Model-Based Tracking using Silhouette Features

J.-B. De La Rivière¹

P. Guitton²

^{1,2} LaBRI*, INRIA FUTURS

351, Cours de la Libération
33405 Talence Cedex

¹ leproux@labri.fr, ² guitton@labri.fr

Résumé

La mise au point de techniques d'interaction performantes constitue un des défis majeurs de la Réalité Virtuelle aujourd'hui. Aussi, nous souhaitons explorer de nouvelles pistes en nous intéressant au pilotage d'applications par des gestes reconnus par capture vidéo. Dans ce but, nous implémentons une méthode de suivi en temps réel de la pose d'un objet articulé (la main ou le corps), basé sur une séquence d'images binaires. Cet article présente notre algorithme dont les caractéristiques sont d'être basé modèle, de reposer sur des primitives de type silhouette, ou encore d'intégrer des contraintes sur les valeurs des paramètres autant que sur leur ordre d'évaluation. Des séquences mono ou stéréo illustrent nos résultats obtenus par analyse du recouvrement des silhouettes d'un modèle 3D et de l'objet suivi, que ce soit une main ou un corps humain.

Mots Clef

Temps-réel, silhouette, pose de la main ou du corps, monoscopique ou multi-caméras, modèle 3D à facettes, hiérarchie de paramètres.

Abstract

Performant interaction techniques are the strongest need of the VR domain. Thus, we choose to implement a real-time video tracking system which extracts hand or body pose from a binary video sequence. Characteristics of our algorithm include the model-based property, the use of silhouettes, and the integration of constraints both on parameters values and parameters evaluation order. Results are shown through mono or stereo sequences. They are obtained by analysing the matching between the 3D model's silhouette and the input one, be the tracked object a human hand or a body.

Keywords

Real-time, silhouette, hand or body posture, mono or stereo cameras, polygonal model, parameters hierarchy.

1 Introduction

L'application de systèmes de suivi vidéo au domaine de la Réalité Virtuelle se révèle très prometteuse. Ils pourraient permettre de s'affranchir de contraintes récurrentes, et de participer à l'immersion des utilisateurs. En effet, alors que les propositions concernant la partie visualisation d'un système de réalité virtuelle ne cessent de s'améliorer, le développement de la partie interaction reste aujourd'hui le problème majeur de ce domaine. Pourtant, l'association d'algorithmes permettant le repérage de la main ou du corps humain à des tâches d'interaction est encore marginal. Les travaux les plus avancés dans cette direction présentent des résultats encourageants, mais ont pour inconvénient de n'être développés qu'en réponse à un besoin caractéristique : pointage dans une CAVE [1], navigation par orientation de la main dans un Reality Center [2], ou encore découpage en zones de mouvements [3]. Ces systèmes, fondés sur une analyse de l'image, partagent l'utilisation d'algorithmes *basés apparence*. A l'inverse, les systèmes dits *basés modèles* utilisent un modèle de description (constitué d'un maillage dans [4] par exemple, ou encore de primitives squelettiques dans [5]) permettant de définir certaines propriétés de l'objet suivi. Ils semblent offrir des résultats de bas niveau, précis, qui pourraient être la base d'interactions différentes, à la manière d'un gant de données. Bien que beaucoup plus coûteux que les premiers, et très stricts quant à l'environnement auquel ils s'appliquent, nous pensons que leurs avantages potentiels justifient de se pencher sur leur utilisation pour extraire des informations destinées à l'interaction.

Nous désirons nous placer par la suite dans un environnement de Réalité Virtuelle de type grand écran. Ceci implique certaines contraintes et libertés, différentes de celles exigées par les méthodes basées modèles proposées à ce

*Université Bordeaux I, CNRS, ENSEIRB

jour [6]. Il nous est donc nécessaire de proposer un nouvel algorithme :

- temps réel, pour éviter les problèmes de latence lors d’interactions (dans l’idéal, fréquence supérieure à 20 Hz) ;
- se satisfaisant d’images binaires à faible résolution ;
- robuste aux occultations ;
- sans restriction quant au nombre de caméras.

Alors que nous présentons dans [6] les premiers résultats respectant ces contraintes, nous détaillons ici nos plus récentes avancées. Dans un premier temps, le paragraphe 2 listera chacun de nos choix d’implémentation concernant l’algorithme de suivi. La partie 3 présentera les différents résultats que nous obtenons ; nous les commenterons de manière globale en 4.

2 Proposition de caractéristiques basées modèle

Les méthodes dites “basées modèle” ont en commun le fait de chercher à minimiser une erreur d’association entre un modèle de description et les données issues d’une capture vidéo. L’erreur est fonction de certaines primitives dans l’image, et représente une mesure de ressemblance entre modèle et objet filmé. Une méthode d’optimisation va influencer sur la pose courante du modèle pour maximiser ce recalage.

2.1 Primitives dans l’image

Les bouts de doigts sont une des primitives classiques utilisées dans la littérature ([7], [5]). Cependant, se reposer sur cette primitive présente l’inconvénient de ne pouvoir assurer un suivi robuste lors de l’occultation du bout d’un doigt. Les arêtes sont une autre source d’informations intéressante ([4], [8]). Mais nos hypothèses de faible résolution et d’image binarisée ne peuvent permettre leur utilisation, puisqu’en conséquence les contours sont amenés à varier fortement d’une image à l’autre. En outre, les contours inter-doigts obtenus en filmant une main doigts tendus et rapprochés ne nous sont pas accessibles.

La silhouette répond quant à elle à nos besoins. Combinée à une mesure d’erreur égale au nombre de pixels différents entre l’image en entrée et l’image synthétisée, elle permet de prendre en compte la seule information que les images binarisées nous permettent d’exploiter : une mesure de recouvrement. Une telle primitive permet de réduire l’impact des problèmes de bruit, chaque pixel ayant une importance égale. Par ailleurs, l’information d’occultation lui est liée implicitement, puisque si un doigt vient à être occulté alors les pixels de la silhouette qui lui appartiennent disparaissent.

2.2 Entrée

Les images que nous utilisons en entrée sont des images binarisées provenant d’une ou plusieurs caméras. Nous n’avons pas choisi de fixer un nombre de caméras minimum ou maximum. Cependant, il est évident que cer-

taines poses ne pourront être correctement extraites si l’algorithme ne possède pas de vue suffisante de l’objet. Dans le même ordre d’idées, nous n’imposons aucune règle sur le placement des caméras. Mais pour que l’ajout d’une caméra ait un intérêt, il faut bien entendu qu’elle soit placée judicieusement, de manière à offrir une vue nouvelle qui résolve certaines occultations de l’objet.

L’étape de binarisation conserve dans l’image seulement la silhouette de l’objet suivi, et supprime le fond. En règle générale, nous travaillons sur des images de résolution faible, de l’ordre de 160x120. Cette résolution correspond à la résolution maximale non-interpolée pour notre webcam. C’est aussi un quart d’une image de 640x480, ce qui pourrait correspondre approximativement à la portion qu’occupe une main filmée dans un espace légèrement restreint. Rien n’empêche l’utilisation de résolutions plus importantes, mais nos performances sont étroitement liées au nombre de pixels. La contrainte de fonctionnement temps réel impose donc une résolution raisonnable.

2.3 Modèles

Il nous faut des modèles assez détaillés pour que la comparaison de silhouettes réelles et synthétiques ait une signification. C’est pourquoi nous avons choisi d’utiliser des modèles 3D polygonaux. Chaque partie rigide est constituée d’un ensemble de triangles qui lui sont propres. A cette géométrie est associé un ensemble de paramètres modélisant les différents degrés de liberté de l’objet articulé.



FIG. 1 – Différentes vues de notre modèle de main

Dans le cas de la main, nous utilisons le modèle 3D de Mojtaba Hosseini, composé de 1500 triangles. Il autorise 21 degrés de liberté¹, 6 ddl étant définis comme globaux (3 ddl pour la position, 3 ddl pour l’orientation). Chaque doigt offre 3 ddl : 1 ddl correspond aux abductions/adductions (utilisé sur la figure 3), les 2 autres codent les flexions de la première et seconde phalange. L’angle déterminant la flexion de la troisième phalange est égal aux deux tiers de celui représentant la flexion précédente (voir par exemple [9] pour une brève description de la modélisation de la main). Le pouce est pour l’instant modélisé sur le principe des autres doigts. La paume est représentée par une unique partie rigide. Notre modèle de corps humain autorise 28 ddl. Parmi eux, 6 sont globaux, 2 fois trois déterminent l’orientation de la tête et du tronc. Les 2 jambes et 2 bras sont ensuite représentés suivant le même schéma : 3 ddl sont associés à l’orientation des bras ou des cuisses, et 1 ddl à la flexion des avant-bras ou du tibia.

¹que nous noterons *ddl*

Nous avons lié deux types de contraintes aux paramètres. Dans un premier temps, leurs valeurs possibles peuvent être restreintes à un domaine prédéterminé, comme dans le cas de l'angle de flexion d'un doigt. L'autre type de contraintes vient du fait qu'il nous sera utile de connaître l'ensemble de paramètres qui affecte la position dans l'image de chaque partie rigide. Par exemple, dans le cas de la main, la position de la première phalange de l'index est dépendante uniquement des 6 ddl globaux et des 2 ddl associés à cette première phalange. La position de l'avant-bras gauche sera fonction uniquement des 6 ddl globaux, des 3 ddl du tronc, des 3 ddl du bras gauche, et de l'unique ddl associé à l'avant-bras gauche. Ces relations peuvent être schématisées sous forme d'un arbre hiérarchique associé à chaque modèle (figure 2).

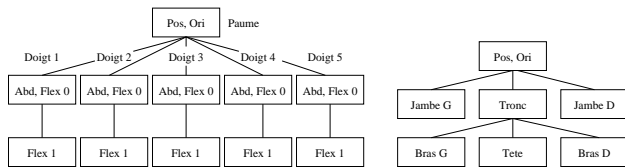


FIG. 2 – Hiérarchie des paramètres pour la main et le corps

Toutes ces données nous permettent de connaître la structure de l'objet articulé qu'on va suivre, et donc ses possibilités de déformation. Or, il nous faut encore générer à partir d'une pose du modèle l'ensemble des images obtenues par sa projection dans les plans correspondant à chacune des caméras. Tout d'abord, il est nécessaire d'associer à chaque caméra réelle une caméra virtuelle, dont les caractéristiques doivent être les plus proches possibles. Ensuite, nous avons choisi d'utiliser un moteur de rendu matériel programmé en OpenGL. En effet, nous considérons que la complexité géométrique plutôt importante de nos modèles nécessite un traitement efficace des opérations de projections comme de transformations, rendu possible par l'exploitation par OpenGL des capacités des cartes vidéos récentes. En pratique, nous lisons dans le framebuffer OpenGL les images correspondant aux projections du modèle dans chacune des caméras.

2.4 Méthode d'optimisation

La contrainte de temps réel a un impact sur l'implémentation de la minimisation de l'erreur. En effet, il est nécessaire que la méthode choisie permette de tirer parti de la moindre analyse qu'on lui laisse conduire. Il est nécessaire que même après un nombre d'itérations réduit, le modèle ait convergé vers la pose correcte. Ce sont ces raisons qui nous ont amené à implémenter une méthode d'optimisation basée sur l'incrémentalité itérative des paramètres pour les faire converger vers la configuration correspondant à un minimum local.

Pour déterminer la pose d'un ddl, nous commençons par déterminer dans quelle direction sa valeur doit être modifiée. En pratique, nous testons lequel de l'incrémenta-

tion ou de la décrémentation de la valeur du ddl réduit le plus l'erreur de recouvrement. A chaque itération suivante, nous continuons de modifier le paramètre dans ce sens tant que l'erreur décroît. Déterminer la pose de l'objet reprend ce principe. Cependant, l'ordre dans lequel les paramètres sont évalués est excessivement important. Par exemple, chercher à analyser l'effet d'une flexion de l'index sur le recouvrement des silhouettes n'a aucun sens si la paume du modèle n'est pas située à la position correcte. Notre analyse itérative se base donc sur les arbres hiérarchiques définis figure 2 : chaque niveau de l'arbre va être traité l'un après l'autre, en partant de la racine. Pour un niveau, chaque paramètre du niveau est itéré si il n'a pas encore été évalué ou si son analyse peut encore permettre une décrémentation de l'erreur globale. Quand tous les ddl liés au niveau courant ont été évalués, les paramètres du niveau suivant sont considérés. Lorsque tous les niveaux ont été complétés, et si il reste encore du temps avant l'arrivée de la nouvelle image, une analyse est relancée en repartant de la racine, ce qui permet de corriger certaines variations hâtives.

Etendre notre mesure d'erreur à plusieurs caméras n'est *a priori* pas trivial. Nos nombreux essais nous ont pourtant montré que définir la mesure d'erreur associée à n couples entrée / projection comme la somme des n erreurs associées à chacune des caméras offre les meilleurs résultats, totalement en adéquation avec le gain espéré lors de l'ajout de caméras.

2.5 Etape d'initialisation

Pour que la silhouette du modèle soit la plus proche possible de celle de l'objet filmé, il est nécessaire d'adapter certaines des caractéristiques géométriques du modèle. Ainsi nous commençons tout suivi par une étape d'initialisation qui va extraire deux types d'informations :

1. extraction de la pose initiale, revenant à déterminer une approximation de la position, de l'orientation et des angles d'abduction de la main présentée dans la position de départ ;
2. extraction des caractéristiques géométriques, soit par exemple la taille de la paume ou la longueur des doigts. Cette partie est menée à bien en utilisant l'algorithme de maximisation du recouvrement décrit plus haut, mais dans lequel les paramètres évalués correspondent aux mesures géométriques recherchées.

3 Résultats

3.1 Présentation

A l'heure actuelle, nous avons implémenté un tel algorithme pouvant suivre soit un corps humain soit une main. La phase d'initialisation n'a pas encore été développée pour le corps, mais fonctionne de manière automatique sur la main. Pour cette raison, nos tests n'incluent que des séquences synthétiques pour le corps humain. Ces résultats ont été obtenus sur un PC équipé d'un Pentium 4 à 2.6 GHz

et d'une Geforce 256. Les 150 projections par image d'une séquence en 80x60 à 10 Hz peuvent donner une idée de la vitesse de notre algorithme. La caméra est une webcam Philips ToUCam Pro.

En règle générale, les séquences sont présentées par paires, avec les images en entrée sur la gauche, et les images résultats de la pose courante du modèle à droite. Sauf si le contraire est précisé, les séquences ont été traitées à 10 images par seconde, ce qui laisse 100 ms d'écart entre une trame et sa suivante. Les numéros des trames dans la séquence sont en général donnés, de façon à permettre une appréciation grossière de la vitesse de déplacement.

3.2 Séquences

Une première séquence (figure 3) montre le suivi de l'abduction des doigts de la main. Cette séquence ne présente pas de difficulté particulière, si ce n'est celle liée à une vitesse élevée. Ces résultats sont très positifs, puisque le modèle suit à la trame près les mouvements de la main en entrée.

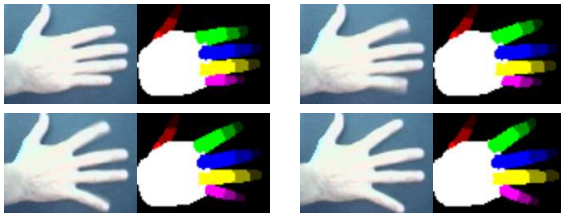


FIG. 3 – Abductions, trames 241, 242, 243, 244.

Les images de la figure 4, suite de la séquence précédente, font intervenir quatre flexions. Là encore les résultats sont bons, avec une vitesse de mouvement assez importante. Notons qu'il arrive, même à faible vitesse, qu'un angle de flexion soit mal évalué quand un doigt se déplie. En effet, l'occultation créée par la paume ne permet plus de vue suffisante sur ce doigt, d'où éventuellement une mauvaise extraction de l'angle d'abduction à ce moment. Alors, il est probable que cette mauvaise abduction perturbe l'axe dans lequel se déplie le doigt du modèle, conduisant à une mauvaise extraction de la flexion du doigt en entrée. Ce problème peut être réglé par l'intégration d'une contrainte sur l'abduction d'un doigt en fonction de la flexion de sa première phalange.

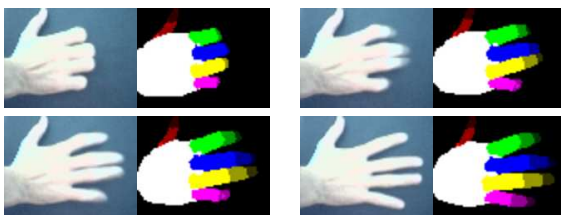


FIG. 4 – Flexions, trames 297, 299, 300, 302.

La figure 5 montre le comportement de notre système après que la main ait subi une rotation. Le taux de rafraîchissement a été fixé à 5 Hz, pour avoir tout le temps nécessaire pour corriger des évaluations fausses pendant la rotation. L'occultation ici est extrême, puisque tous les doigts se projettent à l'origine sur les mêmes pixels. C'est pourquoi on peut voir que l'algorithme baisse le mauvais doigt, mais tente à tout prix de recouvrir la silhouette en entrée.

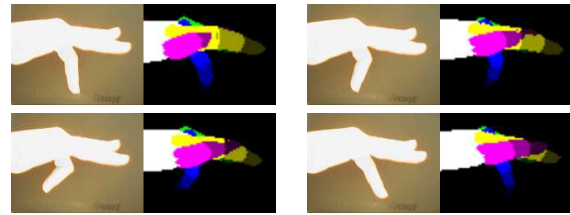


FIG. 5 – Vue après rotation globale, trames 360, 370, 380, 390.

Pour régler le problème de ces occultations, notre implémentation est capable de gérer les flux de plusieurs caméras. Un exemple de deux analyses, l'une avec une caméra et l'autre avec deux est montrée sur la figure 6.

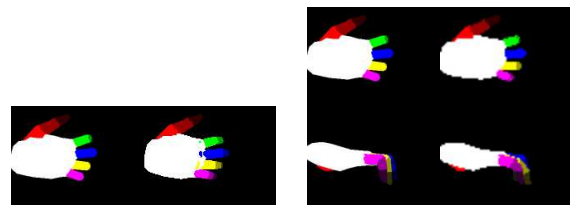


FIG. 6 – Flexions suivie par 1 caméra, par 2 caméras.

La figure 7 illustre la même différence, mais dans le cas du corps humain. Dans les deux images, les silhouettes se recouvrent correctement. Seulement, comme on peut s'y attendre, l'introduction de caméras supplémentaires permet d'éviter certaines erreurs de recalage (par exemple, figure 6, la flexion en entrée de la deuxième phalange est résolue par la flexion de la première phalange du modèle, ce qui est incorrect).

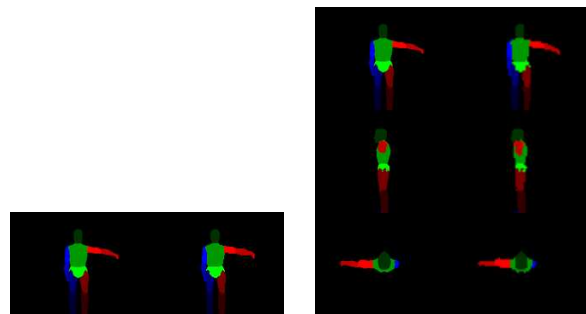


FIG. 7 – Suivi du bras par 1 caméra et 3 caméras.

Nous avons choisi comme derniers exemples deux extraits montrant le résultat d'une occultation temporaire de certains membres. Sur la figure 8, nous pouvons voir que l'occultation des bouts de doigts ne pose aucun problème. Bien entendu, avec la seule vue dont nous disposons ici, nous ne pouvons dire quel doigt est au-dessus de l'autre.

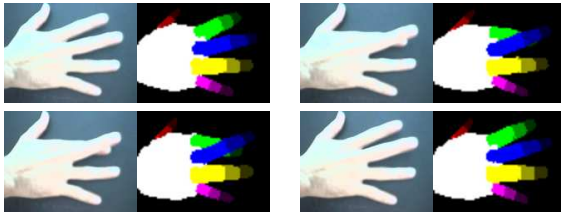


FIG. 8 – Occultation partielle, trames 167, 170, 186, 189.

L'utilisation d'un nombre plus élevé de caméras nous permet alors une meilleure précision, comme le montre la figure 9. Ici, pour passer de la trame 70 à la trame 120, les bras apparaissent souvent l'un devant l'autre. Les résultats semblent corrects, et le modèle est dans une bonne position même dans les caméras où les bras sont fréquemment occultés.

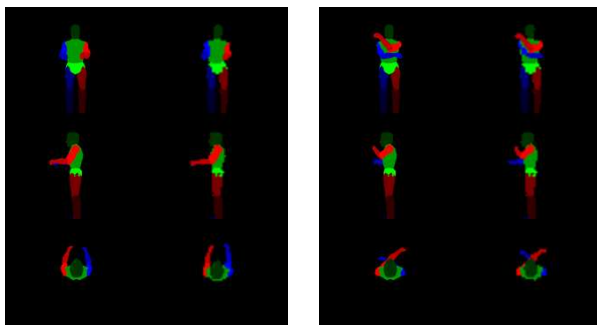


FIG. 9 – Occultations multiples, trames 70 et 120.

3.3 Précision des résultats

Dans le cas de séquences synthétiques, il est possible de comparer les valeurs des ddl extraites par l'algorithme de suivi avec les valeurs effectives dans la séquence. Nous avons donc repris les mesures correspondant à la trame de la figure 7. Nous comparons sur la figure 10 les angles extraits avec 1 et 3 caméras. En ordonnées sont mises ces valeurs dans les 3 configurations, pour chaque paramètre des bras gauche et droit².

²les valeurs représentent respectivement les 3 angles du bras gauche, les 3 angles du bras droit, l'angle de l'avant bras-gauche et celui de l'avant-bras droit.

Angles (degrees)

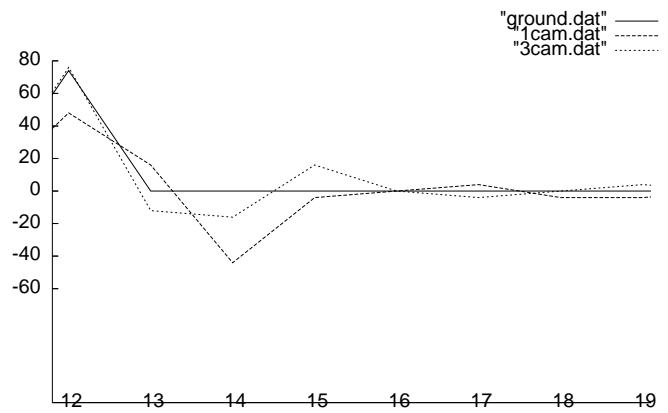


FIG. 10 – Comparaison des angles réels (ground.dat) et des résultats obtenus avec 1 caméra (1cam.dat) et 3 caméras (3cam.dat).

4 Synthèse

4.1 Points positifs

Nous pouvons résumer les résultats précédents en affirmant qu'ils sont très satisfaisants, puisque la silhouette du modèle colle bien à celle en entrée. La vitesse de mouvement reste raisonnable, pour un suivi au moins proche du temps réel : 10 Hz. Ce rafraichissement peut être accéléré, 20 Hz donnant des résultats aussi corrects sur les séquences figures 3 et 4 par exemple.

La gestion des occultations, problème récurrent sur certains systèmes, est ici implicite. Une unique caméra peut être suffisante, cependant elle ne permet plus de gérer certains cas particuliers pour lesquels d'autres points de vue sont nécessaires.

Finalement, nous pensons que ces résultats positifs sont dûs en grande partie au détail de notre modèle 3D (cas de la main). Or, ce détail lui-même est accessible grâce au moteur 3D matériel qui permet un rendu rapide.

4.2 Points négatifs

Bien que répondant à nos attentes, notre système présente certains inconvénients que le recul nous a permis d'analyser. Tout d'abord, il faut noter que, à cause de notre arbre d'évaluation, l'extraction correcte de la pose d'un ddl nécessite obligatoirement une extraction aussi correcte de tous les ddl parentaux dans l'arbre. Ensuite, la majorité des échecs de suivi vient du fait qu'un mouvement trop brusque est filmé. Dans ce cas, le modèle peut alors se recalculer dans une position minimisant localement le non-recouvrement, mais incorrecte. La suite du suivi peut en être fatalement perturbée. Ces décrochages mettent à jour le problème lié à l'absence d'une phase qui permettrait la reprise du suivi dans des conditions à nouveau bonnes. Mais son implémentation est toujours possible ; au pire, cette phase de re-

prise pourrait recquerir le déplacement en pose d'initialisation quand un décrochage est détecté.

Enfin, il ne nous faut pas perdre de vue que toutes nos réflexions sont basées sur la binarité des images. Or, la segmentation d'images filmées est un problème majeur qui ne nous garantit pas l'exactitude de la silhouette extraite. Cependant, nos différents tests ont montré que l'utilisation de la silhouette, au contraire des arêtes, rend notre système plus robuste à une faible perturbation des frontières.

4.3 Perspectives

Outre d'éventuelles modifications ponctuelles des modèles, la prochaine évolution majeure de notre système pourrait être l'intégration d'une étape d'analyse d'image postérieure à chaque nouvelle évaluation. Par exemple, l'image des différences entre images filmées précédente et courante permettrait de relever les paramètres immobiles pour économiser le temps de calcul lié à leur évaluation. Alors que notre moteur 3D semble avoir atteint ses limites, la méthode d'optimisation utilisée jusqu'ici peut certainement être rendue plus efficace. Au minimum, intégrer les évaluations précédentes des paramètres devrait accélérer la convergence vers une pose correcte.

En parallèle, des besoins annexes seront à couvrir. Pour que le système soit facilement utilisable, la phase d'initialisation devra devenir plus rapide et transparente. Il serait particulièrement intéressant d'y réfléchir à nouveau dans l'optique de faciliter une reprise du suivi après décrochage de l'algorithme. De même, le problème de calibration des caméras ne devra pas être négligé : l'association efficace des pixels d'une caméra à l'autre en dépend.

Enfin, reste à appliquer ce système à notre salle de Réalité Virtuelle. Ces choix de silhouette et d'algorithme basé modèle nous permettent d'envisager positivement la gestion d'occlusions entre une personne et un objet ou une autre personne. Cependant, la taille de notre environnement ne permettra sans doute pas une capture de mouvements dans tout l'espace libre. Ainsi, nous nous pencherons sur la meilleure manière de proposer des *zones* d'interaction. Nous serons alors à même de gérer des régions au fond et à l'éclairage contrôlé dans lesquelles le suivi de mouvements pourra être conduit. Finalement, nos résultats sur la vitesse de fonctionnement du système nous semblent encourageants. En effet, ces installations de Réalité Virtuelle disposent d'une puissance de calcul importante. Alors que pour l'instant nous utilisons un PC standard, les équipements spécifiques à la Réalité Virtuelle peuvent nous permettre d'améliorer nos performances. Dans le même ordre d'idée, la parallélisation de notre algorithme est facilement envisageable grâce à la structure hiérarchique qui permet de gérer en parallèle différentes branches de l'arbre.

5 Conclusion

Jusqu'à ce jour, nous avons eu l'occasion d'analyser diverses séquences synthétiques ou réelles. Maintenant, nous nous posons en utilisateurs de notre algorithme, avec en

tête le but de créer des techniques d'interaction à partir des informations qui en sont extraites. Alors que de nombreuses optimisations liées à l'analyse décrite dans cet article verront probablement le jour, alors que beaucoup de développements (calibration des caméras, segmentation robuste des silhouettes) seront nécessaires avant d'atteindre la possibilité d'une interaction, nous sommes toujours convaincus du bienfondé du choix d'un suivi basé modèle qui respecte nos contraintes.

Références

- [1] Moeslund, T., Störring, M., Granum, E. : Vision-based user interface for interacting with a virtual environment. In : conf. DANKOMB. (2000)
- [2] Sato, Y., Saito, M., Koike, H. : Real-time input of 3d hand pose and gestures of a user's hand and its applications for hci. In : Virtual Reality Conference. (2001) p.79
- [3] Davis, J., Bobick, A. : The representation and recognition of action using temporal templates. In : IEEE Conf. of Computer Vision and Pattern Recognition. (1997) 928-934
- [4] Heap, T., Hogg, D. : Towards 3d hand tracking using a deformable model. In : Conference on Automatic Face and Gesture Recognition. (1996) 140-145
- [5] Rehg, J., Kanade, T. : Digiteyes : Vision-based hand tracking for human computer interaction. In : Workshop on Motion of Non-Rigid and Articulated Objects. (1994) 16-22
- [6] de la Rivière, J.B., Guitton, P. : Hand posture recognition in large displays vr environments. In : Int. Gesture Workshop. (2003)
- [7] Wu, Y., Huang, T. : Capturing articulated human hand motion : A divide-and-conquer approach. In : International Conference on Computer Vision. (1999) 606-611
- [8] Ouhaddi, H., Horain, P., Mikolajczyk, K. : Modélisation et suivi de la main. In : Compression et Représentation des Signaux Audiovisuels. (1998) 109-114
- [9] McDonald, J., Toro, J., Alkoby, K., Berthiaume, A., Carter, R., Chomwong, P., Christopher, J., Davidson, M., Furst, J., Konie, B., Lancaster, G., Roychoudhuri, L., Sedgwick, E., Tomuro, N., Wolfe, R. : An improved articulated model of the human hand. The Visual Computer **17** (2001) 158-166