Recalage sur les contours et recalage sur les régions pour l'acquisition 3D des gestes en temps réel par vision monoscopique

David Antonio Gómez Jáuregui, Patrick Horain

▶ **To cite this version:**

David Antonio Gómez Jáuregui, Patrick Horain. Recalage sur les contours et recalage sur les régions pour l'acquisition 3D des gestes en temps réel par vision monoscopique. ORASIS'09 - Congrès des jeunes chercheurs en vision par ordinateur, 2009, Trégastel, France, France. inria-00404618

HAL Id: inria-00404618
https://inria.hal.science/inria-00404618

Submitted on 16 Jul 2009

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

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

Recalage sur les contours et recalage sur les régions pour l'acquisition 3D des gestes en temps réel par vision monoscopique

Region-based *vs.* Edge-based registration for 3D motion capture by real time monoscopic vision

David Antonio Gómez Jáuregui Patrick Horain

Institut TELECOM ; TELECOM & Management SudParis

Dép. EPH, 9 rue Charles Fourier, 91011 EVRY Cedex, France
David.Gomez@IT-SudParis.eu, Patrick.Horain@IT-SudParis.eu

Résumé

Nous nous intéressons à l'acquisition 3D des gestes humains par vision monoscopique en temps réel sans marqueurs. Notre approche procède par recalage d'un modèle 3D articulé du corps sur une séquence vidéo qui consiste à rechercher itérativement la position du modèle et les angles d'articulation qui maximisent la correspondance entre des caractéristiques du modèle 3D projeté et des primitives de l'image. Nous avons précédemment décrit une mise en œuvre à la cadence vidéo d'un recalage initial sur les régions colorées suivi d'un recalage plus précis sur les contours [7]. *Dans ce travail, nous comparons expérimentalement l'erreur résiduelle en fonction du temps de calcul pour chacune de ces primitives de recalage et nous proposons un compromis en fonction de la puissance de calcul disponible.*

Mots clef

Acquisition 3D des gestes, vision monoscopique, recalage sur les régions, recalage sur les contours, recalage 3D/2D.

Abstract

We focus on 3D human motion capture by real-time monocular vision without markers. Our approach works by registering an articulated 3D body model on a video by iteratively the projected 3D model features and the image features with respect to the model position and joint angles. We have previously described a video frame-rate implementation of color-based registration followed by an edge-based registration. In this work, we experimentally compare the residual error vs. *the computation time for each of these features and we propose a compromise depending on computing power available.*

Keywords

3D motion capture, monocular vision, region matching, edges matching, 3D/2D registration.

1 Introduction

Acquérir des gestes par vision artificielle est utile pour un grand nombre d'applications : interfaces homme–machine, animation, interaction avec environnements virtuels, vidéosurveillance, jeux, etc. Nous nous intéressons ici à l'acquisition 3D des gestes par vision monoscopique sans marqueurs [8] en temps réel. Ce problème est rendu difficile par le grand nombre de degrés de liberté du corps humain, les ambigüités dues à l'absence d'information de profondeur, les occultations des parties du corps entre elles et les variations morphologiques et vestimentaires des personnes observées [14].

Nous avons précédemment développé une méthode d'acquisition 3D en temps réel des gestes qui consiste à recaler un modèle 3D articulé du corps humain sur des séquences vidéo en maximisant les correspondances entre des régions colorées de l'image et celles du modèle projeté, puis en minimisant la distance entre les contours de l'image et les contours occultants du modèle 3D [7]. Dans ce travail, nous présentons une analyse expérimentale détaillée de la précision obtenue par rapport au temps de calcul nécessaire pour chacune de ces primitives.

Dans la partie qui suit, nous présentons l'état de l'art de l'acquisition 3D des gestes par vision artificielle. Dans la 3ème partie, nous introduisons notre approche du recalage par mise en correspondance entre les régions colorées puis entre contours. Nos expérimentations de caractérisation de la performance et les résultats obtenus sont présentés dans la partie 4. Finalement, dans la 5ème partie, nous concluons et nous discutons la façon dont un équilibre peut être trouvé dans l'utilisation de ces primitives de recalage tout en tenant compte de la ressource de calcul disponible qui varie selon la plateforme.

2 Travaux antérieurs pour l'acquisition 3D des gestes

Dans l'état de l'art, les approches pour l'acquisition 3D des gestes peuvent être regroupées en deux catégories selon qu'elles utilisent ou non un modèle 3D [14].
La première utilise un modèle 3D du corps humain et recherche la pose qui correspond le mieux à l'image, c'est-à-dire qui minimise une certaine fonction de coût d'association [8]. Des travaux essaient de détecter des parties du corps humain pour estimer la pose 3D en utilisant des contraintes de proximité ou physiques [3]. La cohérence temporelle peut être exploitée entre images successives en suivant une ou plusieurs hypothèses de pose, en particulier au moyen d'un algorithme de filtrage particulaire [6], [17]. Récemment, une modélisation probabiliste a été proposée pour apprendre un geste et ses variations et guider de suivi du mouvement [18], puis a été mise en œuvre dans un espace latent de basse dimension associé à l'espace des poses [13].
Les approches sans modèle n'utilisent pas un modèle 3D du corps humain, mais essaient de déduire directement la pose 3D à partir des images. Elles peuvent reposer sur l'apprentissage d'une correspondance entre les images acquises et la pose 3D [1]. Elles peuvent aussi éviter cet apprentissage en stockant dans une base de données une collection d'exemples de poses 3D et de descripteurs d'image pour adresser la base d'images en interpolant des poses candidates afin de trouver la pose 3D similaire à l'image d'entrée [11].
Les approches précédentes utilisent des primitives telles que la couleur [10], [6], [7], les contours [6], [7], [17], la forme [1], [11], et le mouvement [17], [9].

3 Notre approche

Le recalage sur les régions ou les contours est couramment utilisé pour le suivi d'objets [15], [16]. Nous recalons un modèle 3D articulé de la moitié supérieure du corps humain sur des séquences vidéo [8], [10]. La pose de ce modèle 3D (figure 2a) est décrite par 3 paramètres de position globale du corps et 20 angles des articulations de la partie supérieure du corps (buste, bras, avant-bras, mains, cou et tête). Des primitives (régions, contours) sont extraites pour chaque image capturée d'une part et de la projection du modèle 3D d'autre part. Notre processus de recalage consiste à mettre en correspondance ces primitives de façon optimale par un recalage sur les régions puis par un recalage sur les contours [7] (figure 1).

3.1 Ajustement du modèle 3D et initialisation de la pose

Pour un meilleur recalage sur les vidéos, il convient d'ajuster la morphologie de notre modèle 3D à celle de la personne observée. Ceci est réalisé en ajustant manuellement la pose du modèle et les dimensions (longueur, largeur et hauteur) des différentes parties du

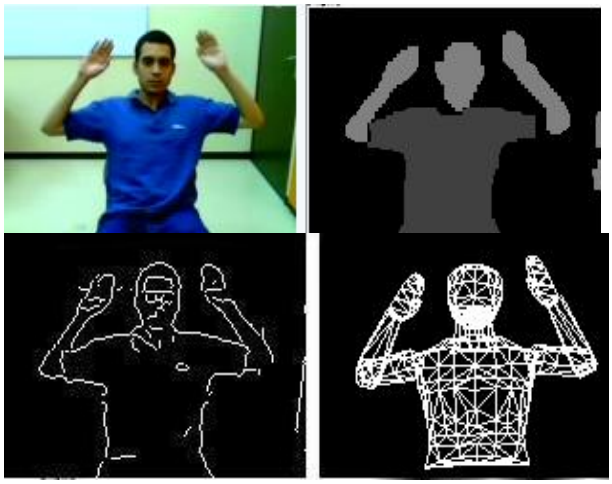

Figure 1: *Notre approche pour l'acquisition 3D des gestes. Les images sont respectivement : l'image acquise, l'image segmentée en classes de couleur, les contours dans le masque d'avant-plan et finalement la projection du modèle 3D du corps humain recalé.*

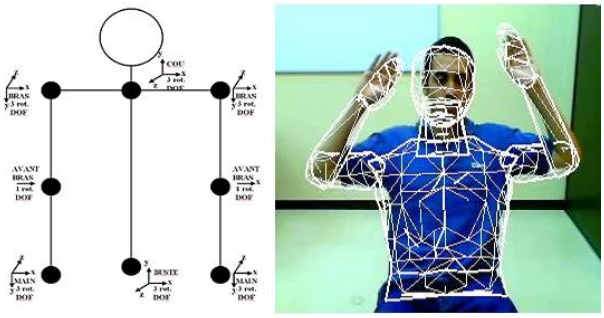

Figure 2 : *Notre modèle 3D de la moitié supérieure du corps humain avec ses 20 degrés de liberté, puis ce modèle maillé superposé à l'acteur pour ajuster la morphologie du modèle sur une image initiale.*

modèle 3D projeté sur une image initiale de la personne observée (figure 2).

3.2 Recalage sur les régions

La silhouette humaine est détectée par différence entre l'image capturée et une image de référence de l'arrière-plan (sans la personne). La silhouette (l'avant-plan) est segmentée en deux classes de couleur (peau et vêtement, ici supposé de couleur uniforme). Les échantillons des couleurs sont extraits automatiquement de la première image. Un échantillon de la couleur de la peau est acquis dans la région du visage détectée par un détecteur de visage Adaboost [19]. Un échantillon du vêtement est acquis en dessous du visage. Ces échantillons sont ensuite modélisés par des distributions gaussiennes dans l'espace colorimétrique HSV. Le modèle 3D, dont chaque partie est munie d'un numéro de couleur (peau ou vêtement), est placé dans la pose décrite par le vecteur de paramètres puis projeté (en utilisant OpenGL [20]) dans le plan de

l'image avec un rendu à plat des numéros de couleur. La correspondance entre la projection du modèle 3D et l'image vidéo segmentée est évaluée par un taux de non recouvrement entre régions colorées :

$$F(q) = \prod_{c=1}^{m} \left(\frac{|A_c \cup B_c(q)| - |A_c \cap B_c(q)|}{|A_c \cup B_c(q)|} \right)^{\frac{1}{m}} \quad [1]$$

où q représente le vecteur des paramètres qui décrivent la posture candidate, A_c est l'ensemble des pixels dans la $c^{ème}$ classe de couleur dans l'image vidéo segmentée, $B_c(q)$ est l'ensemble des pixels dans la $c^{ème}$ classe de couleur dans la projection du modèle, m est le nombre de classes de couleur et $|X|$ représente le nombre de pixels dans X. Cette fonction est ensuite itérativement minimisée par rapport à q en utilisant un algorithme de descente de simplex [12], tout en respectant des contraintes biomécaniques. Plus de détails peuvent être trouvés dans [10]. Cette méthode est robuste car ne nécessite qu'un recouvrement partiel entre régions colorées pour converger. Toutefois, elle n'est pas précise car les pixels de la frontière des régions sont peu nombreux par rapport aux pixels de l'intérieur de la région (figure 3).

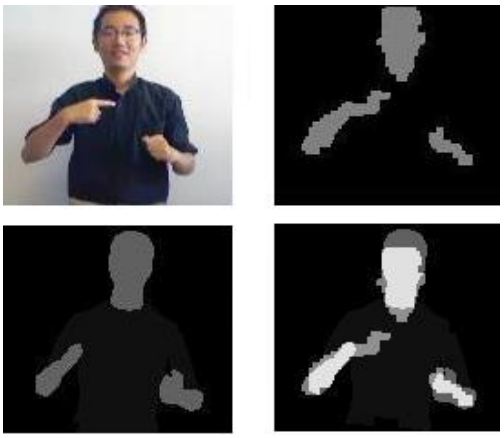

Figure 3 : *Précision limitée du recalage sur les régions : les images sont respectivement l'image acquise, puis segmentée, la projection du modèle 3D et enfin la superposition de la projection du modèle avec l'image segmentée. La pose du modèle 3D diffère de celle de l'acteur observé car le recalage sur les régions n'est pas précis.*

3.3 Recalage sur les contours

Afin d'augmenter la précision, nous utilisons une étape de recalage sur les contours qui consiste en mettre en correspondance les contours de l'image avec les contours occultants du modèle 3D en minimisant la distance qui les sépare [9], [17]. L'état initial du modèle 3D pour cette étape est l'état final du recalage sur les régions. Les contours dans l'image d'entrée sont extraits par un filtre de Deriche [4] uniquement dans la région d'avant-plan. Une carte de distance aux contours est ensuite calculée par un algorithme de chanfrein [2].

Les contours occultants du modèle 3D sont formés des points de la surface où la direction d'observation est tangente à la surface 3D [5]. Ils peuvent être extraits simplement et efficacement en utilisant le *culling* de l'interface OpenGL. D'abord, les polygones du maillage orientés vers l'avant de l'observateur sont projetés avec leurs arêtes et dessinés avec une couleur différente du fond de l'image (les polygones orientés vers l'observateur sont éliminés). Ensuite, l'intérieur de ceux strictement orientés vers l'observateur est rempli avec la couleur de fond pendant les polygones orientés vers l'avant sont ignorés. Ainsi, seules les arêtes occultantes restent marquées différemment du fond. La distance résiduelle entre les contours occultants du modèle projeté et les contours extraits de l'image vidéo est la moyenne de la carte de distance masquée par l'image binaire des contours occultants :

$$D_C = \frac{1}{N_p} \sum_i I_{DT}(p_i) \quad [2]$$

où D_c est la distance moyenne entre contours, I_{DT} est la carte de distance, p_i sont les pixels de la projection des contours occultants du modèle 3D. Le recalage sur les contours consiste à minimiser cette distance entre contours (figure 4) par l'algorithme de descente de simplexe [12] déjà utilisé précédemment.

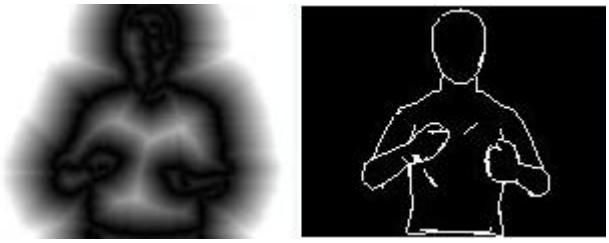

Figure 4 : *Distance entre contours. Les images sont respectivement : la carte de distance aux contours et les contours occultants du modèle 3D.*

3.4 Recalage sur les régions puis sur les contours

Le recalage sur les régions permet d'initialiser le processus du recalage car il est plus robuste, et que le recalage sur les contours permet ensuite d'augmenter la précision (figure 5).

Afin d'évaluer ce gain de précision, nous avons comparé expérimentalement la distance résiduelle entre contours après chacune des étapes du recalage (sur les régions et sur les contours) en itérant jusqu'à la convergence. Nos résultats (figure 6) montrent que le recalage sur les contours permet de corriger certaines erreurs de recalages sur les régions qui apparaissent comme des pics de la distance résiduelle entre contours.

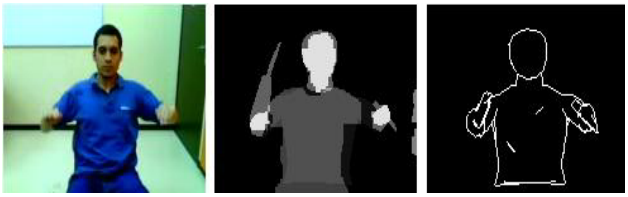

Figure 5 : *Correction du recalage sur les régions. Les images sont respectivement l'image acquise ; le modèle superposé sur l'image segmentée, avec un recalage incorrect du bras droit ; les contours occultants du modèle 3D dans la position finale avec recalage sur les contours.*

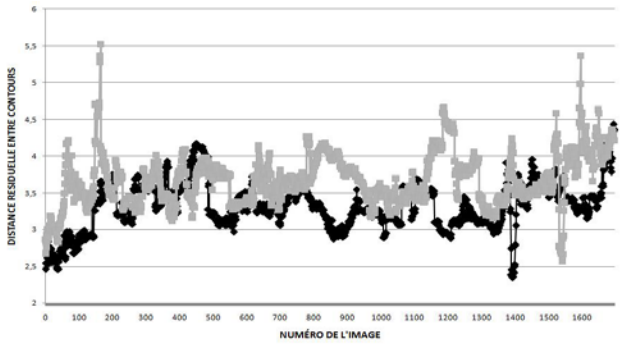

Figure 6 : *Distance résiduelle entre contours obtenue par le recalage sur les régions (en gris) et recalage sur les contours (en noir). Les recalages incorrects apparaissent comme des pics de la distance résiduelle entre contours.*

4 Le processus d'optimisation

L'optimisation dans un espace de paramètres de grande dimensionnalité (23 paramètres dans notre cas) nécessite habituellement un grand nombre d'itérations pour converger. Parce que nous somme intéressé à l'acquisition des gestes en temps-réel, nous devons limiter le temps de calcul et par conséquent, le nombre d'itérations pour chaque image. Malheureusement, ceci dégrade sensiblement la précision du recalage (figure 7).

Le temps de calcul disponible à la cadence vidéo pour chaque image varie avec le nombre d'itérations, la vitesse du processeur (CPU) et la carte graphique (GPU). Il doit être reparti entre l'étape de recalage sur les régions et celle de recalage sur les contours. Nous avons donc analysé expérimentalement la précision obtenue et le temps de calcul consommé en fonction du nombre d'itérations effectuées à chacune de ces étapes afin de déterminer expérimentalement une répartition optimale du temps de calcul.

Nous avons utilisé 6 séquences vidéo[1] présentant des gestes avec occultations, des mouvements rapides, ainsi que des mouvements dans la direction de la profondeur (figure 8) et une personne légèrement de coté.

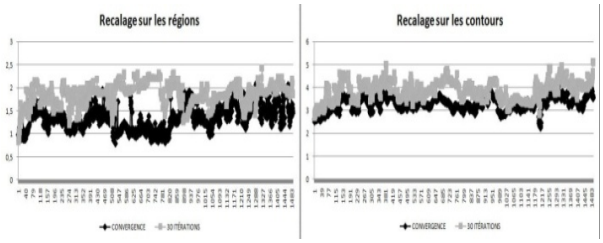

Figure 7 : *Effet d'un nombre limité d'itérations sur l'erreur résiduelle (les ordonnées). En abscisse : le numéro de l'image dans la séquence vidéo. En ordonnée : le taux de non recouvrement minimisé par le recalage sur les régions (à gauche) et la distance moyenne résiduelle entre contours minimisé par le recalage sur les contours (à droite). La line noir correspond à la convergence tandis que la ligne grise correspond à 30 itérations.*

Figure 8 : *Les séquences vidéo utilisées dans nos expérimentations. Les 3 premières séquences vidéo (ligne supérieure) contiennent des gestes avec des mouvements rapides et des occultations. La séquence 4 (en bas à gauche) contient principalement des gestes où l'acteur croise les bras. Dans la séquence 5, la personne n'est pas directement face à la caméra. Dans la dernière séquence, l'acteur tourne sur lui-même.*

[1] Ces séquences vidéo de résolution de 160 × 120 pixels proviennent d'une webcam Logitech QuickCam Pro 5000.

Les tableaux 1 et 2 montrent comment le temps de calcul de notre prototype varie selon la puissance de la plate-forme, le nombre d'itérations et la taille des images. Le tableau 1 montre le temps de calcul sur trois plateformes de calcul différentes[2] en fonction du nombre total d'itérations réparties à égalité entre les deux étapes du recalage. Le tableau 2 permet de comparer le temps de calcul[3] de notre prototype pour l'acquisition 3D des gestes pour des images de plus haute définition.

Tableau 1 : *Temps de calcul sur trois plateformes différentes en fonction du nombre total d'itérations réparties à égalité entre les deux étapes du recalage.*

Nombre d'itérations	Plateforme 1	Plateforme 2	Plateforme 3
40	20 ± 3 ms	22 ± 7 ms	24 ± 6 ms
100	31 ± 5 ms	36 ± 7 ms	37 ± 6 ms
200	46 ± 6 ms	58 ± 7 ms	59 ± 7 ms
300	62 ± 8 ms	79 ± 7 ms	79 ± 7 ms
400	75 ± 9 ms	87 ± 7 ms	95 ± 9 ms
500	93 ± 10 ms	101 ± 10 ms	114 ± 10 ms

Tableau 2 : *Temps de calcul moyen pour des images de plus haute résolution. Les traitements d'image intègrent la suppression de l'arrière-plan, la segmentation de la couleur, l'extraction des contours et le calcul de la carte de distance. Le temps de calcul est comparable pour les itérations de chacune des étapes (recouvrement entre régions et distance entre contours).*

Résolution de l'image	Traitement de l'image	Taux de non recouvrement	Distance entre contours
160 × 120	66 ms	0,3 ms	0,3 ms
256 × 256	137 ms	0,6 ms	0,5 ms
320 × 240	149 ms	0,7 ms	0,6 ms
480 × 480	482 ms	1,9 ms	1,5 ms
512 × 512	560 ms	2,2 ms	1,7 ms
640 × 480	600 ms	2,5 ms	1,9 ms

4.1 Initialisation du recalage sur les contours

Le nombre d'itérations étant limité, il importe d'initialiser l'étape du recalage sur les contours avec un simplex réduit. Le recalage sur les contours est initialisé en utilisant le simplex final de l'étape du recalage sur les régions. De cette manière, le recalage sur les contours commence à chercher dans un espace réduit autour du résultat l'étape précédente dont nous avons vu qu'elle fournit une estimation imprécise mais robuste de l'optimum recherché. Nous pouvons observer les résultats expérimentaux de cette solution proposée dans la figure 9.

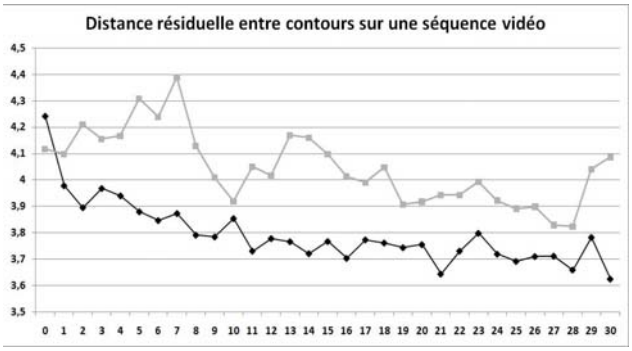

Figure 9 : *Distance résiduelle entre contours sur une séquence vidéo. Le nombre d'itérations sur les contours figure en abscisses d'un total de 30 itérations (les itérations supplémentaires sont réalisées par le recalage sur les régions). La ligne noire est l'erreur résiduelle en utilisant le simplex final du recalage sur les régions pour initialiser le recalage sur les contours. La ligne grise est l'erreur résiduelle en utilisant un grand simplex initial, qui ne permet pas d'atteindre une petite distance entre contours.*

4.2 Evaluation des performances

La précision du recalage peut être analysée à partir des valeurs résiduelles du taux de non-recouvrement des régions et de la distance entre contours, ainsi que le nombre de décrochages en fonction du nombre d'itérations effectuées.
Nous avons analysé la performance de 1 à 500 itérations, pour des temps de calcul inférieurs à 100 millisecondes (voir le tableau 1), compatibles avec une acquisition à 10 Hz ou plus. Nous avons mesuré les valeurs résiduelles du taux de non recouvrement et de la distance entre contours moyennes pour toutes les images de chaque séquence vidéo. Nous comptons aussi les décrochages, où les valeurs résiduelles présentent des pics supérieurs à un seuil défini. Nous montrons les résultats expérimentaux pour la séquence vidéo 2 dans les figures 10 à 13.

[2] Plateforme 1 : CPU Intel Core 2 Extreme Q9300 2.53GHz avec un GPU NVIDIA Quadro FX 3700M.
Plateforme 2 : CPU Intel Pentium 4 3,6 GHz et GPU NVIDIA Quadro FX 1400.
Plateforme 3 : CPU Intel Pentium 4 3,0 GHz et GPU NVIDIA GeForce 9600 GT.
[3] CPU Intel Core 2 Extreme Q9300 2.53GHz avec un GPU NVIDIA Quadro FX 3700M.

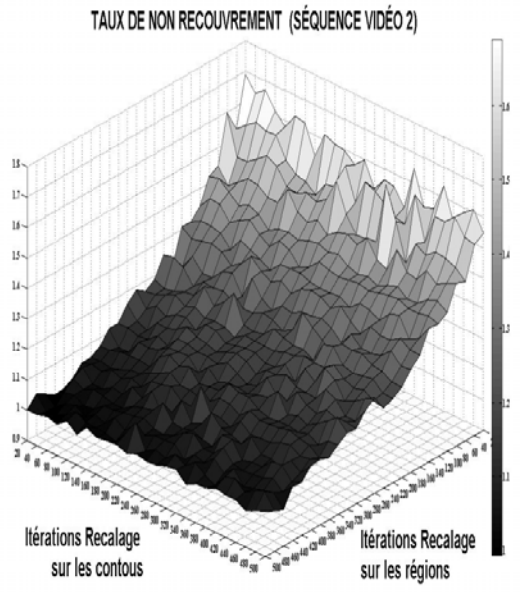

Figure 10: Erreur moyenne résiduelle du taux de non recouvrement (l'axe vertical) par rapport aux nombre d'itérations de recalage sur les régions et de recalage sur les contours (axes horizontaux) obtenue sur la séquence vidéo 2. Les expérimentations sur les séquences vidéo 1, 3, 4, 5 et 6 donnent des résultats similaires.

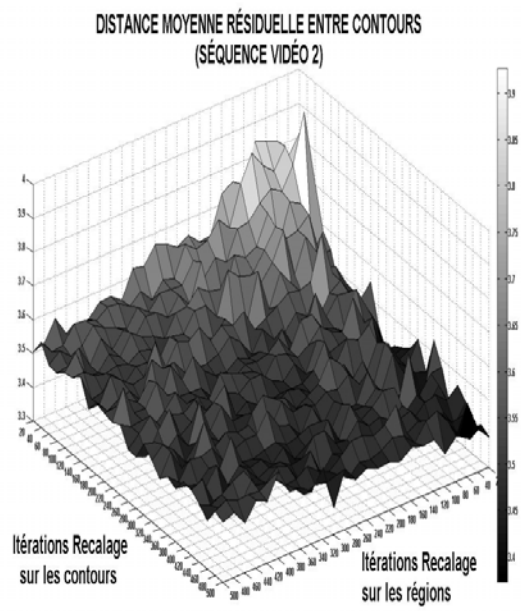

Figure 11: Erreur moyenne résiduelle de la distance entre contours (l'axe vertical) par rapport au nombre d'itérations de recalage sur les régions et de recalage sur les contours (axes horizontaux) obtenue sur la séquence vidéo 2. Les expérimentations sur les séquences vidéo 1, 3, 4, 5 et 6 donnent des résultats similaires.

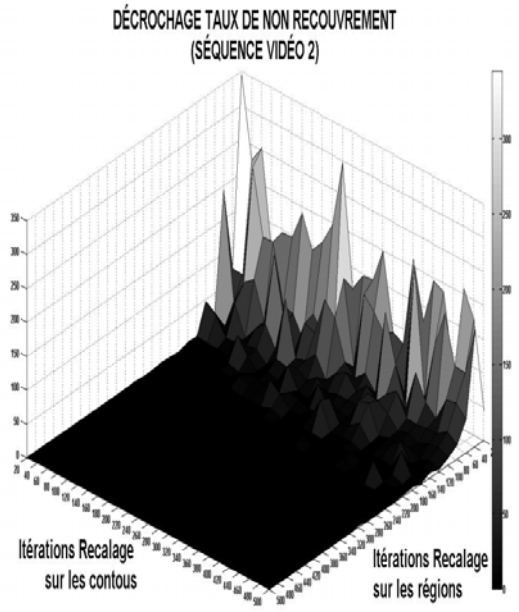

Figure 12: Nombre de décrochages du taux de non recouvrement (l'axe vertical) par rapport au nombre d'itérations de recalage sur les régions et de recalage sur les contours (axes horizontaux) obtenue sur la séquence vidéo 2. Les expérimentations sur les séquences vidéo 1, 3, 4, 5 et 6 donnent des résultats similaires.

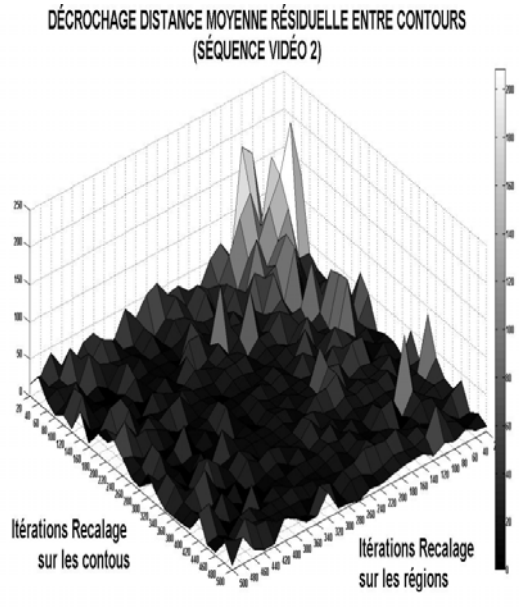

Figure 13: Nombre de décrochages de la distance moyenne résiduelle entre contours (l'axe vertical) par rapport au nombre d'itérations de recalage sur les régions et de recalage sur les contours (axes horizontaux) obtenue sur la séquence vidéo 2. Les expérimentations sur les séquences vidéo 1, 3, 4, 5 et 6 donnent des résultats similaires.

4.3 Analyse des résultats

À partir des figures 10 et 11, nous constatons que le recalage sur les régions arrive plus rapidement à la convergence que le recalage sur les contours. Les figures 12 et 13 montrent que le recalage sur les contours est moins stable (grand nombre de pics) que le recalage sur les régions. Nous devons donc combiner la robustesse et la stabilité du recalage sur les régions et la précision du recalage sur les contours.

Pour atteindre le temps réel, nous devons limiter le nombre d'itérations en fonction de la puissance de calcul de la plateforme. Le nombre d'itérations possible est mesuré expérimentalement pour chaque plateforme (tableau 1). Pour avoir la meilleure performance, nous donnons la priorité à la stabilité du recalage lorsque le nombre d'itérations est inferieur à 200 (valeur choisie expérimentalement à partir de la figure 12) en consacrant toutes les itérations au recalage sur les régions. Au-delà, le nombre de décrochage du recalage sur les régions devient relativement petit (figure 12), ce qui permet d'améliorer la précision du recalage par des itérations supplémentaires de minimisation de la distance entre les contours (figures 5 et 6).

Notons que, si ces performances (figures 10 à 13) sont comparables pour toutes les séquences vidéo testées, la séquence vidéo 6 présente le plus grand nombre de décrochages à cause de l'ambigüité occasionné par les mouvements de rotation vers la profondeur (l'acteur tourne sur lui-même) en images monoculaires.

5 Conclusions

Nous avons évalué un algorithme pour l'acquisition 3D des gestes par vision monoscopique en temps-réel par recalage d'un modèle 3D articulé et analysé expérimentalement les performances d'une approche par mise en correspondance de régions colorées et d'une approche par mise en correspondance de contours lorsque le nombre d'itérations est limitée par une contrainte de temps réel. Nous avons mis en évidence la robustesse et la stabilité du recalage sur les régions et la précision du recalage sur les contours, et avons proposé un compromis en fonction du nombre total d'itérations possible en temps réel sur la plateforme de calcul utilisée.

Bibliographie

[1] A. Agarwal, B. Triggs, Recovering 3D human pose from monocular image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, pp. 44-58, 2006.

[2] G. Borgefors, Distance transformations in digital images, *Computer Vision, Graphics and Image processing*, Vol. 34, pp. 344-371, 1986.

[3] G. Cheung, S. Baker, T. Kanade, Shape-from-silhouette for articulated objects and its use for human body kinematics estimation and motion capture, *Computer vision and pattern recognition*, Madison, Wisconsin, USA, pp. 16-22, 2003.

[4] R. Deriche, Fast algorithms for low-level vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 78-87, 1990.

[5], J.-S. Franco, E. Boyer, Une approche hybride pour calculer l'enveloppe visuelle d'objets complexes, *ORASIS'03*, Gérardmer, pp. 67-74, 2003.

[6] M. Fontmarty, F. Lerasle, P. Danes, Data Fusion within a modified Annealed Particle Filter dedicated to Human Motion Capture, *IEEE / RSJ International Conference on Intelligent Robots and Systems IROS 2007*, San Diego, CA, USA, pp. 3391-3396, 2007.

[7] D. A. Gómez Jáuregui, P. Horain, F. Baroud, Acquisition 3D des gestes par vision monoscopique en temps réel. *Conférence MajecSTIC'08*, Marseille, France, 2008.

[8] P. Horain, M. Bomb, 3D Model Based Gesture Acquisition Using a Single Camera, *Proceedings of IEEE Workshop on Applications of Computer Vision*, Orlando, Florida, December 3-4, pp. 158-162, 2002.

[9] S. Lu, G. Huang, D. Samaras, D. Metaxas, Model-based integration of visual cues for hand tracking, *Proceedings of IEEE workshop on Motion and Video Computing*, Orlando, Florida, pp. 119-124, 2002.

[10] J. Marques Soares, P. Horain, A. Bideau, M.H. Nguyen, Acquisition 3D du geste par vision monoscopique en temps réel et téléprésence, *Actes de l'atelier Acquisition du geste humain par vision artificielle et applications*, Toulouse, pp. 23-27, 2004.

[11] G. Mori, J. Malik, Recovering 3D human body configurations using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, pp. 1052-1062, 2006.

[12] J. A. Nelder, R. Mead, A simplex method for function minimization, *Computer Journal*, Vol. 7, pp. 208-313, 1965.

[13] J. Pang, L. Qing, Q. Huang, S. Jiang, Monocular Tracking 3D People by Gaussian Process Spatio-Temporal Variable Model. *International Conference on Image Processing, ICIP2007*, San Antonio, Texas, USA, Vol. 5, pp. 41-44, 2007.

[14] R. W. Poppe, Vision-based human motion analysis: An Overview, *Computer Vision and Image Understanding*, Vol. 108, pp. 4-18, 2007.

[15] M. Pressigout, E. Marchand, Real-time 3D Model-Based Tracking: Combining Edge and Texture Information. *In IEEE Int. Conf on Robotics and Automation, ICRA'06*, Orlando, Florida, USA, pp. 2726-2731, 2006.

[16] C. Schmaltz, B. Rosenhahn, T. Brox, D. Cremers , J. Weickert, L. Wietzke, G. Sommer, Region-based pose tracking, *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, Girona, Spain, Vol. 4478, pp. 56-63, 2007.

[17] C. Sminchisescu, B. Triggs, Estimating Articulated Human Motion with Covariance Scaled Sampling, *International Journal of Robotics Research*, Vol. 22, pp. 371-393, 2003.

[18] R. Urtasun, D. J. Fleet, P. Fua, 3D people tracking with gaussian process dynamical models, *Proceedings of the Conference on Computer Vision and Pattern Recognition CVPR'06*, New York, NY, Vol. 1, pp. 238-245, 2006.

[19] P. Viola, M. Jones, Rapid Object Detection Using a Boosted Cascade of Simple Features, *IEEE Computer Vision and Pattern Recognition*, Vol. 1, pp. 511-518, 2001.

[20] R. S. Jr. Wright, B. Lipchak, N. Haemel, *OpenGL SuperBible: Comprehensive Tutorial and Reference 4th edition*, Addison-Wesley Professional, Ann Arbor, Michigan, USA, pp. 127-172, 2007.