Acquisition 3D du geste par vision monoscopique en temps réel et téléprésence

3D Gestures Acquisition in Real Time Using a Single Camera and Telepresence

José Marques Soares, Patrick Horain, André Bideau, Manh Hung Nguyen

GET / INT / EPH – Intermédia
9 rue Charles Fourier, 91011 EVRY Cedex, France

Jose_Marques.Soares@int-evry.fr, Patrick.Horain@int-evry.fr, Andre.Bideau@int-evry.fr

Résumé

La vidéo peut être un support pour la communication gestuelle dans les environnements informatiques collaboratifs à distance, au prix d'une charge réseau importante et d'une interface complexe comportant une fenêtre par utilisateur. Des acteurs virtuels permettent de restituer à bas débit dans un monde virtuel les gestes de l'ensemble des utilisateurs distants. Nous présentons un prototype pour l'acquisition 3D des gestes humains par vision monoscopique en temps réel, ainsi que leur restitution par des avatars dans un environnement 3D de type téléprésence pour le partage d'applications.

Mots clef

Acquisition de gestes, vision artificielle, suivi, temps réel, animations d'avatars, réunion virtuelle, téléprésence.

Abstract

Video can support gesture communication in collaborative environments, at the cost of high bandwidth and one window interface per user. Virtual actors allow low bit rate gesture transmission and display in a single virtual world for several users. We describe a prototype system for 3D gestures acquisition in real time using a single camera and the restitution of these gestures by avatars in a 3D environment for telepresence and application sharing.

Keywords

Gesture acquisition, computational vision, tracking, real time, avatar animation, virtual meeting, remote presence.

1. Introduction

Les gestes sont fréquemment employés comme complément de la communication humaine permettant d'améliorer la compréhension. Dans des environnements informatiques pour la collaboration à distance, la vidéo permet de percevoir les gestes des interlocuteurs distants. Toutefois, ceci implique autant de flux vidéo et de fenêtres d'affichage que de participants, ce qui peut poser des problèmes de charge réseau et d'ergonomie de l'interface.

Une alternative consiste à restituer la *communication non verbale* au moyen d'avatars humanoïdes 3D animés [1]. Dans ce cas, des acteurs virtuels restituent à distance et à bas débit les gestes des utilisateurs.

Dans cet article, nous rappelons brièvement notre approche pour l'acquisition des gestes humains en 3D par vision monoscopique et sans marqueur, décrite dans une publication précédente [2]. Puis, nous décrivons une nouvelle mise en œuvre et une adaptation de ces algorithmes pour atteindre le temps réel. Enfin, nous présentons succinctement un exemple d'animation à distance d'acteurs virtuels dans un monde 3D partagé [3] à partir d'une acquisition par « webcam ».

2. L'acquisition des gestes

Notre méthode [2] consiste à recaler un modèle 3D articulé du corps humain sur une séquence vidéo. Nous avons utilisé un modèle 3D de la moitié supérieure du corps humain possédant 23 degrés de liberté. Les images sont segmentées à partir d'une classification grossière des couleurs, et le modèle 3D est coloré selon ces classes. Un taux de non-recouvrement calculé entre l'image segmentée et la projection du modèle 3D coloré est minimisé itérativement par un algorithme de descente de simplexe [4], tout en respectant des contraintes biomécaniques. Ceci permet d'acquérir les gestes de la partie supérieure du corps humain par vision artificielle monoscopique, sans marqueur et sans connaissance *a priori* des gestes attendus.

Cette méthode repose sur une approche par analyse et synthèse très simplifiée, exploitant des silhouettes colorées. La procédure d'optimisation du recalage est très coûteuse car notre fonction de comparaison (taux de non-recouvrement) ne peut être dérivée analytiquement, ce qui interdit des algorithmes rapides d'optimisation de type descente de gradient [5, 6].

En contrepartie, cette méthode offre l'avantage d'être très simple à mettre en œuvre :

- la projection du modèle 3D peut être accélérée au moyen de la carte graphique des PC standards, et
- le jeu d'instructions SIMD des processeurs modernes permet de paralléliser la classification des

couleurs et surtout la procédure d'évaluation du recalage.

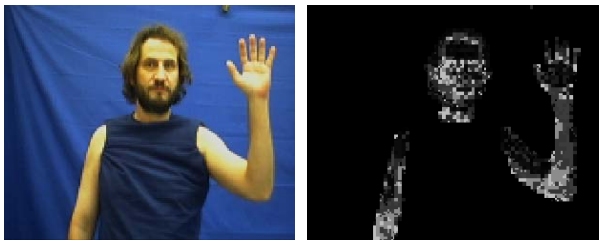
La prochaine section présente la mise en œuvre de ces technologies, ainsi que quelques autres améliorations, pour une acquisition en temps réel.

3. Mise en œuvre en temps réel

3.1. Segmentation des images

La peau et les vêtements, supposés de couleur uniforme, constituent des classes de couleur. Celles-ci peuvent être discriminées par leur teinte, peu sensible aux variations d'éclairage.

A partir d'échantillons de couleur (peau et vêtements) issus d'une image vidéo, on commence par générer les histogrammes de teinte de chaque classe. Pour chaque image de la séquence, et pour chaque classe de couleur, ces histogrammes normalisés sont utilisés pour transcoder les teintes en probabilités d'appartenance aux classes [7]. La Figure 1 montre l'image de probabilités *(b)* créée pour la couleur de la peau à partir de l'image *(a)*.



(a) *(b)*

Figure 1 – Image vidéo (a) et image de probabilité d'appartenance à la couleur de la peau (b).

Chaque pixel est attribué à la classe la plus probable s'il dépasse un seuil, ou est classé comme arrière-plan sinon. Une ouverture morphologique permet de réduire le bruit de la classification.

Cette segmentation est mise en œuvre au moyen de la bibliothèque OpenCV [8] conçue pour la vision artificielle en temps réel et dont le code source est ouvert. Celle-ci offre des traitements performants exploitant les jeux d'instructions étendus MMX et SSE des processeurs Pentium.

3.2. Projection du modèle 3D avec OpenGL

Les articulations du modèle 3D sont organisées selon la hiérarchie standard H-ANIM [9]. On extrait d'un fichier VRML décrivant un humanoïde la géométrie maillée de chaque segment du corps ainsi que les positions des articulations. Chaque segment du modèle est associé à une classe de couleurs.

Le calcul de la projection du modèle 3D coloré dans le plan image (*Figure 2b*) est accéléré en utilisant la carte graphique (dont les PC standards sont désormais habituellement équipés) au moyen de l'interface OpenGL [10]. Pour améliorer les performances, les commandes de restitution des segments rigides du modèle articulé sont mémorisées dans la carte graphique sous forme d'une *liste d'affichage*, et seuls les paramètres articulatoires sont transmis à chaque nouvelle projection.

3.3. Evaluation du recalage

Le recalage optimal est recherché par des différences entre l'image vidéo segmentée et l'image du modèle projeté.

Une posture donnée du modèle 3D est projetée et comparée à l'image vidéo segmentée en examinant les couleurs des pixels des deux images. Notre critère de comparaison est un taux de non-recouvrement [2] :

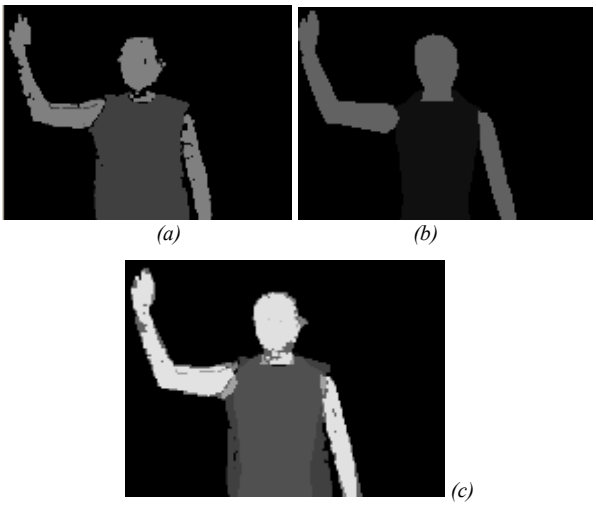
$$F(q)=\prod_{c=1}^{m}\left(\frac{\left|A_c\cup B_c(q)\right|-\left|A_c\cap B_c(q)\right|}{\left|A_c\cup B_c(q)\right|}\right)^{\frac{1}{m}} \quad (1).$$

où q est le vecteur de paramètres articulatoires décrivant 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 la projection des segments du modèle porteurs de la $c^{ème}$ couleur, m est le nombre de classes de couleur (hormis l'arrière-plan) et $|X|$ désigne le nombre de pixels dans un ensemble X.

Ces intersections et réunions entre les ensembles A_c et $B_c(q)$ peuvent être calculées par une comparaison systématique des pixels de l'image vidéo segmentée (*Figure 2a*) et de l'image du modèle projeté (*Figure 2b*). Le coût de ce traitement peut être réduit en combinant ces deux images en une seule puis en extrayant les informations ensemblistes de son histogramme. Pour cela, sous l'hypothèse que nous utilisons moins de 16 classes de couleur, les couleurs de la première image sont codées sur les 4 bits de poids forts et celles de la deuxième image sur les 4 bits de poids faible. Ensuite, une addition des 2 images donne une image de superposition (*Figure 2c*) où la valeur binaire de chaque pixel indique à quelle intersection $A_x \cap B_y(q)$ il appartient, x et y étant parmi les m classes de couleur ou la classe d'arrière-plan. Ces pixels sont comptés sur l'histogramme de l'image résultante. Un ensemble $A_c \cup B_c(q)$ apparaissant dans l'équation (1) est l'union des $A_c \cap B_x(q)$ et $A_y \cap B_c(q)$, x et y décrivant l'ensemble des classes. Confondant les classes de couleurs avec leurs codes hexadécimaux sur 4 bits, le nombre de pixels dans $A_c \cup B_c(q)$ est la somme des cellules suivantes de l'histogramme, où x et y peuvent être n'importe quelle classe de couleur ou arrière-plan :

- cx : intersection avec la classe x dans l'image 2,
- yc : intersection avec la classe y dans l'image 1, $y \neq c$.

Ici aussi, nous utilisons la bibliothèque OpenCV qui offre des fonctions efficaces d'addition des images et de calcul d'histogramme.



(a) *(b)* *(c)*

Figure 2 – (a) Image vidéo segmentée ; (b) modèle projeté ; (c) superposition des 2 images (somme).

3.4. Détection des régions de mouvement

L'optimisation doit être répétée pour chaque image vidéo. Ce processus est très coûteux parce qu'il requiert un grand nombre d'évaluations de postures candidates.

Etant donnée N le nombre de paramètres qui décrivent une posture, l'algorithme de descente du simplexe utilisé doit faire converger N+1 sommets vers un optimum [4].

Pour réduire les calculs, il est possible de détecter les parties du modèle en mouvement et d'ajuster seulement leurs paramètres. Ceci permet de réduire la dimension de l'espace de recherche.

Les régions de mouvement peuvent être détectées à partir des images vidéo segmentées successives, mais il n'est pas possible d'identifier directement les segments du modèle impliqués dans ce mouvement.

Pour résoudre ce problème, nous définissons, sur le modèle, des groupes de segments (buste, bras droit, bras gauche, tête) et nous affectons un numéro de couleur différent pour chacun d'eux. La couleur de la peau est donc maintenant représentée par 3 valeurs différentes, une pour chaque groupe de segments. Nous remplaçons ainsi le précédent modèle 3D coloré par un modèle 3D coloré par groupe de segments.

Une nouvelle image segmentée est superposée (sommée) à la projection du modèle 3D coloré par groupe dans la posture où il a été recalé à l'image précédente. Les cardinaux des intersections des ensembles de pixels sont calculés comme expliqué précédemment. Les groupes de segments dont les cardinaux des intersections ont changé de manière significative entre deux images successives sont considérés comme étant en mouvement. Par exemple, la *Figure 3* montre une importante variation de la région d'intersection du bras droit.

Le processus d'optimisation est ainsi limité aux paramètres qui contrôlent les groupes en mouvement ou les segments qui en dépendent dans la hiérarchie du modèle.

Figure 3 – Superposition du modèle 3D avec 2 images segmentées successives.

3.5. Parallélisme entre la segmentation et l'optimisation

Le recalage du modèle sur une image vidéo segmentée est indépendant de la segmentation de l'image suivante. Ces tâches, qui font respectivement largement appel à la carte graphique pour les projections 3D et au processeur central pour la segmentation, utilisent des ressources de calcul différentes. Elles peuvent donc être accomplies simultanément par 2 séquences de calcul (*threads)* en parallèle.

3.6. Résultats

Nous avons utilisé une webcam Philips ToUcam PRO PCVC640K pour l'acquisition vidéo au format 160 × 120 pixels.

Pour comparer différentes configurations matérielles, nous avons utilisé comme référence une même séquence vidéo enregistrée. L'algorithme de descente du simplexe à été limité à 100 itérations. Nous avons obtenu les résultats suivants:

Configuration		**Images par seconde**
UC	**Carte 3D**	
Intel Pentium IV 1.6 GHz, 256 Mo RAM	ATI Radeon 7500	3
	ATI Radeon 9800	6
	NVIDIA GeForce 3	11
Intel Pentium IV 2.2 GHz, 512 Mo RAM	NVIDIA GeForce 3	12
	NVIDIA GeForce FX 5900	12

L'importance de l'accélération graphique est mise en évidence par la comparaison des résultats concernant le processeur Pentium à 1,6 GHz.

Cependant, avec le processeur 2,2GHz, la carte graphique de hautes performances NVIDIA GeForce FX 5900 ne permet pas des calculs plus rapides que la carte de milieu de gamme GeForce 3. Un logiciel de profilage montre que 90% du temps d'exécution est consacré à l'évaluation du recalage et que les 2/3 de ce temps sont utilisés pour le transfert des données entre la carte graphique et l'unité centrale.

4. Application à la téléprésence

La téléprésence peut fournir aux utilisateurs une plus grande sensation d'immersion dans des environnements déportés que la téléconférence traditionnelle en permettant la manipulation et le contrôle des objets distants [11].

L'acquisition des gestes et leur restitution à distance permettent une valorisation de la communication dans les environnements de téléprésence tout en utilisant une transmission de données en bas débit.

Nous avons utilisé le programme d'acquisition de gestes en temps réel décrit précédemment pour animer des avatars dans un environnement virtuel habité [3]. Cet environnement permet le partage d'applications immergées dans un espace 3D où les actions réalisées pour chaque participant sont reproduites sur un tableau virtuel par leurs avatars respectifs (*Figure 4*). Ces avatars, codés suivant le standard H-ANIM [9], peuvent être animés par des gestes calculés par cinématique inverse.

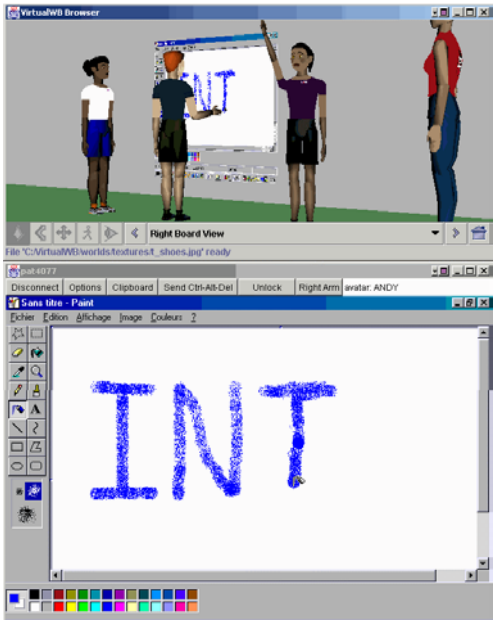


Figure 4 – Environnement pour le travail collaboratif où des actions réalisées sur le document partagé sont simulées par un avatar dans un monde 3D habité.

Dans cet environnement, la sensation d'immersion est améliorée par la possibilité de percevoir les présences, les actions, les sortie et les entrée des utilisateurs dans le monde virtuel 3D affiché sur une fenêtre unique. Cependant, la communication gestuelle entre des participants est limitée à un ensemble d'animations prédéfinies.

Nous avons intégré l'acquisition des gestes et leur restitution à distance dans cet environnement. Les paramètres articulatoires acquis à chaque image vidéo sont convertis au format MPEG-4/BAP [12] et envoyés au serveur d'animations de l'environnement virtuel en utilisant le protocole UDP. Ce serveur les redistribue vers les clients pour l'animation des avatars.

Après cette intégration, la communication par gestes entre les participants peut être établie de façon spontanée dans l'environnement de partage et non pas uniquement au travers d'animations prédéfinies (*Figure 5*).

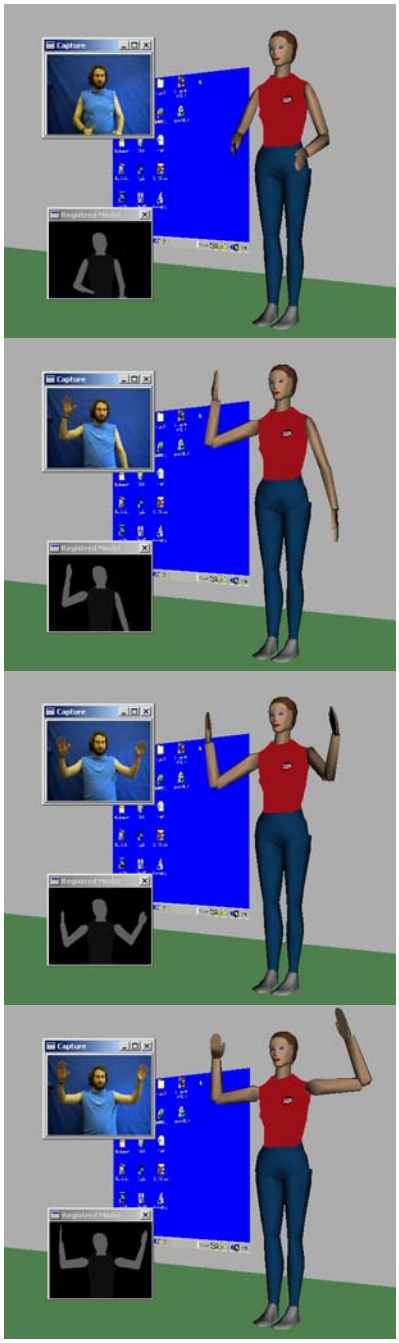


Figure 5 – Animation d'un avatar à partir des gestes acquises en temps réel

5. Conclusion et perspectives

Nous avons développé un prototype pour l'acquisition 3D des gestes humains par vision monoscopique qui étend notre travail précédent [2]. L'utilisation efficiente des

ressources matérielles et logicielles nous a permit d'atteindre le temps réel.
Ce prototype a été intégré à notre environnement virtuel de partage d'applications [3] pour animer des avatars humanoïdes à distance. L'animation est réalisée à partir des gestes acquis en temps réel. Cette intégration permet une communication gestuelle plus naturelle entre des utilisateurs dans un monde virtuel 3D habité.
Nous envisageons d'évaluer cet environnement comme un outil complémentaire dans le cadre d'un projet de formation à distance.

Remerciement. Ce travail a été réalisé avec le soutien financier partiel du gouvernement brésilien au travers du projet CAPES/COFECUB n°266/99-I.

Bibliographie

[1] A. Vuilleme-Guye, T. K. Capin, I. Pandzic, N. Thalman, D. Thalman, "Nonverbal Communication Interface for Collaborative Virtual Environments", *Virtual Reality J.*, 1999, vol. 4, pp. 49-59.

[2] P. Horain, M. Bomb, "Acquisition du geste humain 3D par vision monoscopique" , Actes des 8èmes journées d'études et d'échanges Compression et Représentation des Signaux Audiovisuels (CORESA'03), Lyon, 16-17 janvier 2003, pp. 269-272. www-eph.int-evry.fr/~horain/ARC-LSF/publications.html

[3] J. Marques Soares, P. Horain, A. Bideau, "Sharing and immersing applications in a 3D virtual inhabited world", *Laval Virtual 5th virtual reality international conference (VRIC 2003)*, Laval, France, 13-18 May 2003, pp. 27-31. http://www-eph.int-evry.fr/~soares/WB

[4] J. A. Nelder, R. Mead, "A Simplex Method for Function Minimisation", *Computer Journal,* Vol. 7, 1965, pp. 308-313.

[5] 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, 3-4 December 2002, pp. 118-124.
www.cs.sunysb.edu/~samaras/papers/wmvcf02.pdf.

[6] C. Sminchisescu, B. Triggs, "Estimating Articulated Human Motion with Covariance Scaled Sampling", *to appear in International Journal of Robotics Research,* 2003. http://www.cs.toronto.edu/~crismin/PAPERS/css_ijrr03.pdf .

[7] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface", *Intel Technology Journal, 2nd Quarter*, Intel Corporation, Microcomputer Reseach Lab, Santa Clara, CA, 1998. http://developer.intel.com/technology/itj/q21998/articles/art_2.htm

[8] Intel Corporation, *Open Source Computer Vision Library – OpenCV.*
http://www.intel.com/research/mrl/research/opencv

[9] Humanoid Animation Working Group, *H–ANIM specification.* http://H-Anim.org.

[10] Silicon Graphics Inc., *OpenGL – The Industry's Foundation for High Performance Graphics.* http://www.opengl.org

[11] M. G. Mair, "Telepresence – the technology and its economic and social implications", *IEEE International Symposium on Technology and Society,* 1997, 20-21 June, pp. 118-125.

[12] T. K. Capin, D. Thalmann, "Controlling and Efficient Coding of MPEG-4 Compliant Avatars", *International Workshop on Synthetic-Natural Hybrid Coding and Three-Dimensional Imaging, IWSNHC3DI'99*, Santorini, Greece, 1999.