Communication gestuelle et télévirtualité: Interaction autour d'une application partagée et acquisition des gestes par vision artificielle en temps réel

José Marques Soares, Patrick Horain, André Bideau

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

nat-marques@bol.com.br, Patrick.Horain@int-evry.fr, Andre.Bideau@int-evry.fr

Résumé

Les gestes sont un moyen naturel de communication et peuvent apporter une très grande valeur ajoutée dans le travail collaboratif à distance. La vidéo peut supporter cette communication, mais au prix d'une charge réseau importante et d'une interface complexe comportant une fenêtre par utilisateur. L'animation d'acteurs virtuels permet la communication gestuelle à distance à bas débit en intégrant, dans un environnement unique, tous les éléments participant de la collaboration. Nous présentons un environnement prototype de monde virtuel habité qui permet la communication gestuelle par des avatars soit en simulant des actions réalisées sur l'application partagée, soit en restituant des gestes acquis en temps réel par un système de vision artificielle monoscopique sans marqueur.

Mots clefs

Travail collaboratif, animations d'avatars, acquisition de gestes, télévirtualité.

1 Introduction

Dans le partage d'applications à distance, la perception des actions individuelles et la communication gestuelle entre participants sont souvent limitées. La vidéo peut fournir un support pour la communication gestuelle, au prix d'une charge réseau importante et d'une interface complexe qui nécessite une fenêtre par utilisateur [1].
Des environnements de réalité virtuelle proposent d'améliorer cette perception en représentant chaque utilisateur par un objet virtuel appelé avatar. Habituellement sous la forme humanoïde, ces objets peuvent partager un même espace 3D virtuel présenté dans une fenêtre unique. Les avatars peuvent être animés par des gestes qui augmentent le sens d'immersion dans l'espace de collaboration [2].
Nous présentons un environnement virtuel 3D habité qui améliore la perception des interactions lors du partage d'application dans un environnement collaboratif à distance. Les actions réalisées par un utilisateur sur une application partagée sont reproduites par son avatar dans l'espace virtuel. De plus, les utilisateurs peuvent établir une communication non verbale par des gestes naturels acquis en temps réel par un système de vision artificielle.

2 Collaboration dans un monde virtuel 3D habité

Nous proposons le partage dans un monde 3D d'applications 2D. La vidéo permet difficilement de construire un environnement intégré, capable de valoriser les gestes de communication et de manipulation sur l'application réalisés par des participants distants.
Dans notre environnement, l'activité gestuelle des avatars permet de visualiser les actions réalisées sur l'application 2D.
Le scénario collaboratif rassemble, dans un même espace virtuel, l'objet de partage (l'application d'interface 2D) et les avatars représentant les participants.

2.1 Interface hybride 2D+3D

Le scénario proposé permet d'augmenter le sens de collaboration entre des participants distants, sans modifier la manière d'interagir avec l'application partagée.
Ainsi, nous utilisons une interface hybride 2D+3D [3] composée de deux espaces distincts: *l'espace applicatif* et *l'espace immersif*. Le premier présente, sans dégradation de l'affichage, l'application 2D partagée sur laquelle les utilisateurs interagissent directement. Le partage de cette application est réalisé par l'intermédiaire d'un client VNC (*Virtual Network Computing*) [4] qui permet d'afficher l'interface graphique d'un ordinateur distant. *L'espace immersif* est un monde 3D virtuel multi-utilisateur qui contient un *tableau virtuel* sur lequel est projeté *l'espace applicatif*. Des avatars humanoïdes, définis suivant le standard H-ANIM [5], représentent les utilisateurs qui participent au travail.

2.2 Représentation des interactions

Les événements dans l'*espace applicatif* déclenchent, dans l'espace immersif, des animations en temps réel. L'avatar suit de la main la position associée aux actions réalisées par l'utilisateur sur l'*espace applicatif*. Pour cela, les angles de rotation des articulations du bras sont calculés en temps réel par cinématique inverse.
Un utilisateur peut demander la main pendant qu'un autre interagit avec l'application. Dans ce cas, son avatar lève le bras en indiquant qu'il est en attente (Figure 1). Les utilisateurs peuvent aussi déclencher, pour leurs avatars, à n'importe quel instant, des animations prédéfinies.

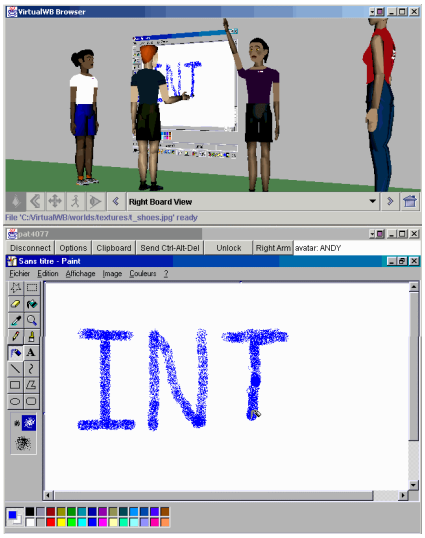


Figure 1 - *Animation par cinématique inverse (les actions de l'utilisateur sur l'application partagée sont reproduites par son avatar).*

Cet environnement peut aussi permettre de rapprocher les mondes réel et virtuel. Les actions d'un participant qui travaille sur un tableau réel, augmenté avec vidéoprojecteur et équipement MIMIO [6], peuvent actuellement être restituées par son avatar dans le monde virtuel partagé (Figure 2).

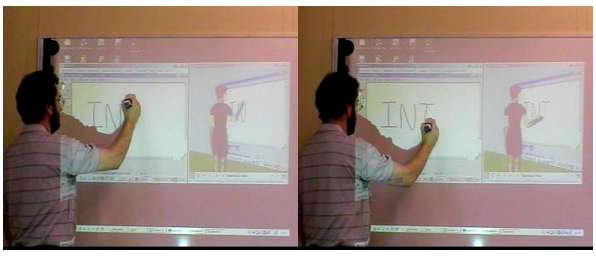


Figure 2 - *Application sur un tableau augmenté.*

2.3 Architecture

Ce prototype a été construit sur une architecture client serveur écrite en Java. Nous utilisons Java3D et Xj3D pour charger et animer des modèles écrits en VRML.

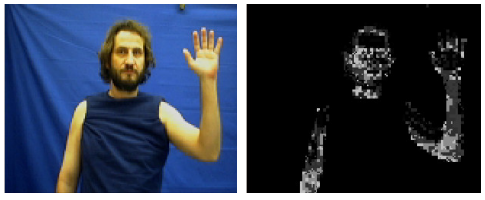
Le calcul des angles des articulations du bras de l'avatar est effectué par cinématique inverse en utilisant la bibliothèque IKAN [7]. Un serveur d'événements contrôle l'accès des clients à l'application partagée et leur diffuse les données nécessaires à l'animation des avatars. Celles-ci sont codées au format MPEG-4/BAP [8].

3 Acquisition et restitution des gestes en temps réel

Pour permettre une animation plus libre des avatars dans le monde virtuel 3D, nous avons développé un système d'acquisition des gestes humains volontairement basée sur des technologies facile à mettre en œuvre. Dans cette optique, nous adoptons une approche par vision monoscopique, sans utilisation de marqueurs [9] et nous utilisons une caméra de type webcam et un PC grand public muni d'une carte graphique standard. La méthode utilisée permet de recaler la moitié supérieure du corps d'un modèle 3D humanoïde sur une séquence vidéo. Les positions des articulations et les géométries maillées de chaque segment de ce modèle sont extraites d'un fichier VRML décrivant un humanoïde selon la hiérarchie standard H-ANIM.

3.1 Extraction des caractéristiques des images vidéo

Chaque image vidéo est segmentée à partir d'une classification en quelques classes de couleur. Les vêtements, supposés de couleur uniforme, et la peau constituent les classes de couleur. Celles-ci sont 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 [10]. La Figure 3 montre l'image de probabilités *(b)* créée pour la couleur de la peau à partir de l'image *(a)*.



(a) *(b)*

Figure 3 - *Image vidéo (a) et image de probabilité d'appartenance à la couleur de la peau (b).*

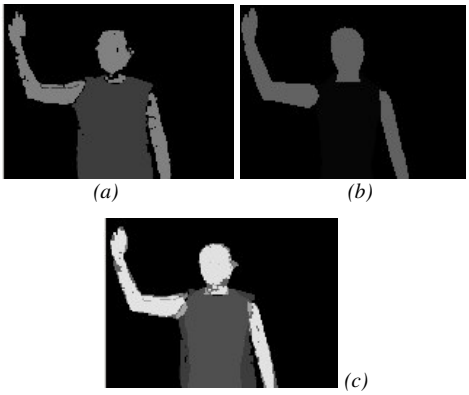
Chaque pixel est attribué à la classe la plus probable s'il dépasse un seuil, ou est classé comme arrière-plan sinon. Une ouverture morphologique permet de réduire le bruit de la classification.

3.2 Recalage 3D/2D

Le recalage optimal est recherché par des différences entre l'image vidéo segmentée et l'image du modèle projeté. Une fonction de coût est minimisée itérativement par un algorithme de descente de simplexe [11], tout en respectant des contraintes biomécaniques. Notre critère de comparaison est un taux de non-recouvrement [9] :

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

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.



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

Figure 4 – *(a) Image vidéo segmentée ; (b) modèle projeté ; (c) superposition des 2 images (somme).*

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 4a) et de l'image du modèle projeté (Figure 4b). 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 4c) 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, $y \neq c$.

3.3 Mise en œuvre en temps réel

Pour atteindre le temps réel, nous avons mis en œuvre une solution qui exploite un processeur Pentium et une carte graphique accélératrice 3D [12]:

- la bibliothèque OpenCV [13] permet de segmenter les images vidéo et d'effectuer les comparaisons d'image nécessaires au calcul du taux de non-recouvrement, pendant l'évaluation de chaque posture candidate. OpenCV offre des traitements performants exploitant les jeux d'instructions étendus MMX et SSE des processeurs Pentium ;
- le calcul de la projection du modèle 3D dans le plan image est accéléré en utilisant la carte graphique (dont les PC standards sont désormais habituellement équipés) au moyen de l'interface OpenGL [14].

Nous traitons jusqu'à 12 images par seconde en utilisant un PC Pentium IV 2,2 GHz, 512 Mo de mémoire vive, avec par exemple une carte graphique NVIDIA GeForce 3 ou GeForce FX 5900. 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. Ceci explique pourquoi l'utilisation d'une carte graphique plus performante ne permet pas des calculs plus rapides.

3.4 Restitution des gestes dans le monde virtuel

Nous avons intégré le système d'acquisition des gestes présenté dans cette section à l'environnement collaboratif présenté à la section 2. Pour cela, les paramètres articulatoires acquis à chaque image vidéo sont convertis au format MPEG-4/BAP [8] et envoyés au serveur d'événements de l'environnement virtuel qui à son tour les redistribue vers les clients connectés à l'environnement.

Cette intégration permet la communication par gestes libres entre les participants dans l'environnement de partage et non pas uniquement au travers d'animations prédéfinies (Figure 5).

4 Conclusions

Nous avons présenté un environnement de télévirtualité [15] pour le partage à distance d'applications 2D. Cet environnement valorise la communication

gestuelle entre les utilisateurs. Ceux-ci sont représentés par des avatars humanoïdes articulés qui peuvent être animés pendant le travail collaboratif. Ces animations permettent d'augmenter le sens de collaboration entre les utilisateurs de deux différentes manières :

- les actions des utilisateurs sur l'application partagée sont reproduites par leurs avatars dans le monde virtuel et repérées visuellement par tous les participants ;
- les gestes naturels peuvent être acquis en temps réel et restitués à distance, en permettant une valorisation de la communication dans l'environnement collaboratif.

La représentation des collaborateurs et de l'application de partage dans un même espace virtuel permet d'atténuer l'effort cognitif des utilisateurs qui peuvent percevoir toutes les actions distantes dans une même fenêtre. De plus, comparativement aux environnements basés sur la vidéoconférence, cette approche permet de réduire la charge réseau.

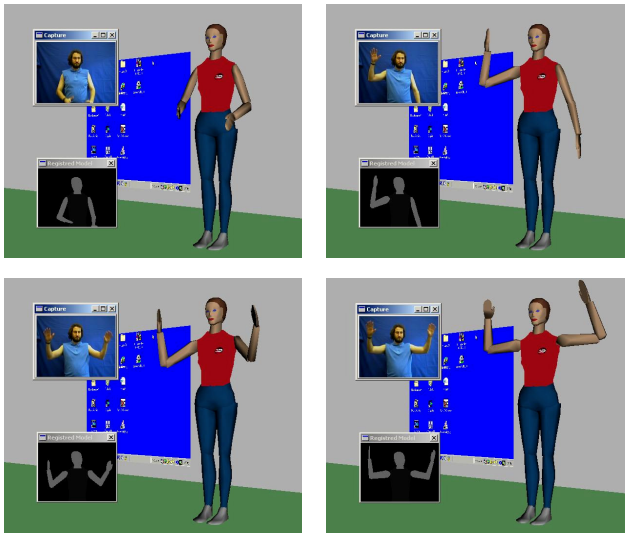


Figure 5 - *Acquisition de gestes en temps réel et restitution à distance par un avatar.*

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

5 Références

[1] W. H. Leung & T. Chen, Creating a Multiuser 3-D Virtual Environment, *IEEE Signal Processing Magazine*, May 2001, pp. 9-16.

[2] 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.

[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/~horain/MarquesSoares

[4] AT&T Laboratories, VNC – Virtual Network Computing. Http://www.uk.research.att.com/vnc

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

[6] Vitual Ink Corporation, MIMIO products, http://www.mimio.com/index.shtml

[7] D. Tolani, A. Goswami, N. Badler, Real-time inverse kinematics techniques for anthropomorphic limbs, *Graphical Models*, 2000.

[8] 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.

[9] P. Horain, M. Bomb, Acquisition du geste humain 3D par vision monoscopique, *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

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

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

[12] J. M. Soares, P. Horain, A. Bideau, M. H. Nguyen, Acquisition 3D du geste par vision monoscopique en temps reel et téléprésence, *Atelier d'Aquisition du geste humain par vision artificielle et applications,* Toulouse, 27 janvier 2004, pp. 23-27
http://www-eph.int-evry.fr/~horain/AtelierGeste/

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

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

[15] P. Quéau, Televirtuality: The merging of telecommunications and virtual reality, *Computers & Graphics, Volume 17*, *Issue 6*, November-December 1993, pp. 691-693.