3D Gesture Acquisition in Real Time
Using a Single Camera and Televirtuality

Patrick Horain, José Marques Soares, André Bideau, Manh Hung Nguyen
GET / INT / EPH
9 rue Charles Fourier, 91011 EVRY Cedex, France
Patrick.Horain@int-evry.fr

Abstract

Gestures in collaborative environments can be conveyed by video, 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-gesture acquisition in real time using a single camera and gesture restitution by avatars in a 3D televirtuality environment to support application sharing.

1. Introduction

Gestures are a complement to human communication. They help understanding.

Multipoint videoconferencing systems allow to perceive gestures by a distant participant. However, using a video stream per participant may significantly increase network traffic and implies an interface with as many windows as participants. As an alternative, transmitting body joints parameters and animating 3D humanoid avatars allow non-verbal communication [1] at a low bit rate. Avatars can be shown in a single virtual environment that may include an application window.

In this paper, we briefly introduce our approach for 3D-gesture acquisition using a single camera, without markers and without *a priori* knowledge, that has been described in a previous publication [2]. Then, we present a new implementation and adaptations for real time processing. Finally, as an example application, we show how gesture acquired with a plain webcam can be remotely reproduced by animating virtual actors in a shared 3D-world [3].

2. Gesture acquisition

Our approach [2] consists in registering an articulated 3D model of the human upper body onto video sequences. The 3D-model has 23 degrees of freedom. It is colored with few colors (skin, clothes…). Video pixels are classified based on their hue, one class per model color, and images are then morphologically filtered and segmented. Model candidate postures are compared with images by computing a non-overlapping ratio between the segmented image and the 3D-colour model projection. This evaluation function is minimized to get the best registration. Biomechanical limits constrain the registered postures to be morphologically reachable. Statistical dynamic constraints enforce physically realistic movements.

This method can be regarded as a very simplified approach of image analysis by synthesis, working only with colored silhouettes.

Optimizing the registration is computationally intensive because our evaluation function cannot be analytically derived. Thus, gradient descent algorithms [4, 5] cannot be used. Instead, we used a downhill simplex algorithm [6] that requires evaluation of a large number of candidate postures. As on order of magnitude, in our experiments involving 23 degrees of freedom, we had to evaluate typically several hundreds postures per image. In counterpart, this evaluation function can be implemented very efficiently:

- 3D-model projections can be accelerated using the graphic card of a standard PC;
- the SIMD instructions supported by modern processors allows fast implementations of color classification and non-overlapping ratio evaluation.

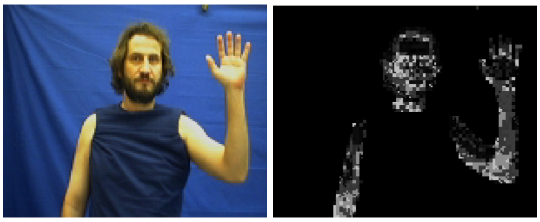
We describe hereafter how we took advantage of these technologies along with some other algorithm improvements for real time processing.

3. Real time acquisition

3.1. Image segmentation

We suppose that skin and clothes have uniform distinct hues and that they can be described as a small set of color classes. Chrominance, and thus hue, is known to be little sensitive to lighting variations.

For each color class, a hue histogram is initially learnt from a sample video image, and a color probability distribution is derived by normalization. For each video image and each color class, an image of the probability a pixel belongs to that class is then computed [7]. Figure 1 shows a probability image for the skin color class (b) corresponding to image (a).



(a) (b)

Figure 1. Video image (a) and skin color probability (b).

Image segmentation is obtained by setting each pixel to the class with highest probability, given it is higher than some *a priori* threshold, or to the background otherwise. A morphological opening allows to reduce noise.

This processing has been implemented using OpenCV [8], a free library developed by Intel for real time computer vision applications. High processing rates are achieved by exploiting the MMX and SSE extended sets of instructions for Pentium processors.

3.2. Projection of the 3D model using OpenGL

The 3D model is organized according to the H-ANIM hierarchy [9]. The articulated segments meshes and the joints positions are extracted from a VRML file describing the humanoid model. Each segment of the model is associated with a color class.

Computing the projection of the 3D colored model in the image plane (Figure 2b) can be accelerated with the graphic card that nowadays most PCs hold. OpenGL [10] is a common standard API. Faster execution is achieved by storing commands for rendering rigid segments into the graphic card as OpenGL display lists. These can then be re-executed for each posture (*i.e.* set of joints parameters) to be evaluated.

3.3. Comparing the model projection with the segmented image

Optimal registration is searched by minimizing the mismatch between the segmented video image (Figure 2a) and the image projection of the 3D-model. Model candidate postures are evaluated against the segmented video images by comparing their color features. For that purpose, we use a non-overlapping ratio [2]:

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

where q is the vector of joints parameters describing a candidate posture, A_c is the set of pixels in the c^{th} color class in segmented image, $B_c(q)$ is the projection of the model segments with the c^{th} color class, m is the number of color classes (background excluded) and $|X|$ designates the number of pixels in set X.

These intersections and unions between sets A_c and $B_c(q)$ can be calculated by systematic comparison of the pixels in the segmented and the projection images for all the color classes c. We accelerated this process by combining the images of A_c and $B_c(q)$ regions into a single image and then counting pixels from its histogram. Under the hypothesis that we use at most 15 colors, the pixels color or background indices in the first (respectively second) image can be stored in the 4 high-order (respectively low-order) bits. Adding these 2 images generates a superposition image (Figure 2c) where the binary value of each pixel indicates to which intersection $A_x \cap B_y(q)$ it belongs, where x and y are a color class or the background. Pixels in all the $A_x \cap B_y(q)$ are then counted by a single histogram. $A_c \cup B_c(q)$ in equation (1) is the union of all the $A_c \cap B_x(q)$ and $A_y \cap B_c(q)$, where x and y can be any color or the background. Confusing color classes and their four bits hexadecimal indices, the number of pixels of $A_c \cup B_c(q)$ is the sum of the following histogram bins where x and y can be any color class or the background:

- cx: intersection with class x in the picture 2,
- yc: intersection with class y in the picture 1, $y \neq c$.

Once again, we take advantage of the efficient SIMD implementation of OpenCV functions for image addition and histogram.

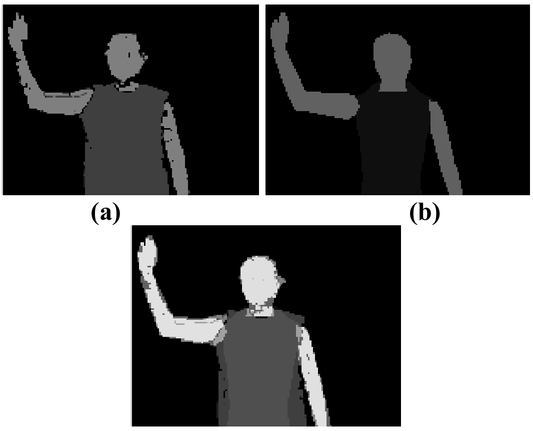


Figure 2. (a) Segmented video image; (b) projected model image; (c) their superposition (sum).

3.4. Detecting the moving regions

The registration optimization process is repeated for each video image. This process is computationally intensive, since it requires a large number of evaluations of candidate postures.

Let N be the number of parameters that describe a posture. The down-hill simplex algorithm we use has to make N+1 vertices converge towards the optimum [6]. So the computation cost is grows with the number of parameters.

To reduce the execution time, it is possible to detect the moving model parts and to adjust only their parameters, so decreasing the parameters space dimension. Moving regions can be detected from the segmented successive video images, but it is not possible to identify directly the model segments implied in this movement.

To solve this problem, we define groups of model segments (bust, right arm, left arm, and head) and we use different indices for the colors of these groups. So, the skin color class is now represented with 3 different index values, one for each group of segments. We replace the previous colored 3D model with a group-colored 3D model.

A new segmented video image is superposed (added) to the image of group-colored 3D model projected in the posture that was registered at previous image. The cardinals of intersections of pixels sets are computed as we explained previously. Groups of segments whose intersections cardinals have significantly changed from one image to the next are considered to be in movement. For example, figure 3 shows a large variation of the image intersection with the right arm group.

The optimization process is then limited to those parameters that control only the moving groups of segments, or that control their children segments in the model hierarchy.

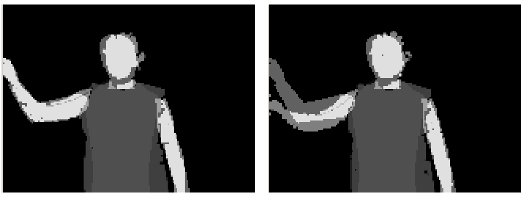


Figure 3. Superposition of the 3D model registered at time *t* with the segmented video images at *t* and *t*+1.

3.5. Parallelism between segmentation and optimization process

Model registration on an image is independent of the segmentation of the next video image. Projection of the 3D model relies on the graphic card. Image segmentation is achieved in the central processor unit. So these tasks use different calculation resources. They can be accomplished in 2 parallel computation threads.

3.6. Results

We grabbed video with a Philips webcam ToUcam PRO PCVC640K at the definition of 160 × 120 pixels.

To compare hardware configurations, we saved a reference video sequence. The down-hill simplex optimization process has been limited to 100 iterations. We got the following results:

Configuration		**Images per second**
Processor and memory	**Graphic Card**	
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

The importance of the graphic card acceleration in the optimization process is emphasized by the comparison with the 1.6 GHz Pentium.

However, with the 2.2 GHz processor, the top performance NVIDIA GeForce FX 5900 graphic card does not perform better than the middle range GeForce 3. Software profiling tools show that the evaluation process takes 90% of CPU time. 2/3 of it are used for data transfer between the graphic card and the central unit. So, this transfer is the main bottleneck that limits the benefit of high-end graphic accelerator cards.

4. Application to televirtuality

Televirtuality consists in representing users in a remote virtual environment. It is the result of hybridization of telecommunications and computer images, exploiting the functional possibilities offered by computer graphics techniques as simulation, gestural interaction and coupling of the body with the image, immersion in "virtual worlds"… [11]

Gesture acquisition and their remote restitution can improve communication in televirtuality environments with low bit rate communication.

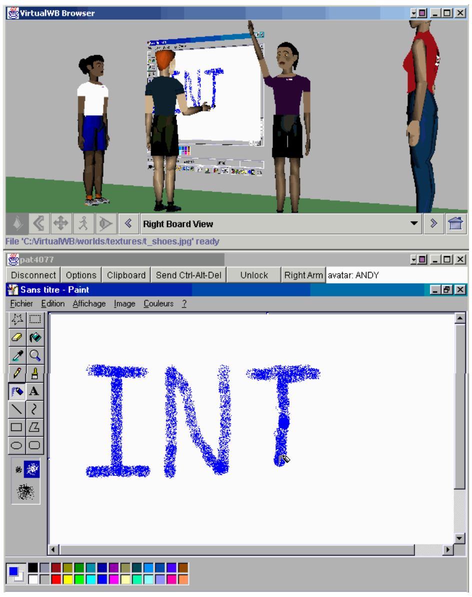


Figure 4. Collaborative environment where actions of participants on a shared application are shown by avatars in a 3D inhabited world.

We use our gesture acquisition system as an input to animate avatars in a virtual inhabited environment we developed previously [3]. This environment allows to share and immerse applications in a 3D space where the actions of participants are reproduced on a virtual board by their respective avatar (Figure 4). These avatars follow the H-Anim standard [9]. They can be animated by inverse kinematics. In this environment, avatars increase the sense of collaboration among co-workers sharing a 2D-application because it is possible to perceive in a single window who is present, who is entering or exiting, and to see users' actions on the shared application. However, communication by gestures is limited to some predefined animations.

We have integrated gestures capture and remote restitution into this environment. The posture parameters acquired for each image have been converted to the MPEG-4/BAP format [12]. They are sent with the User Datagram Protocol (UDP) to the animation server of the virtual environment. This server re-distributes them to clients to animate avatars. This allows participants to communicate by gestures in the shared virtual environment in a more natural way than using predefined animations (Figure 5).

5. Conclusion and perspective

We have developed a prototype for real time 3D gesture acquisition using only one camera. It extends our previous work [2] by an efficient usage of both software and hardware to reach real time. It has been integrated in our collaborative virtual environment [3] to animate avatars with participant's gestures acquired in real time. This allows natural gesture-based communication among users that are shown in a single inhabited virtual world.

We consider evaluating this environment as a complementary tool for distant learning.

Acknowledgement

The Brazilian government supports this work through the project CAPES/COFECUB n°266/99-I.

6. References

[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, "3D Model Based Gesture Acquisition Using a Single Camera", *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV 2002)*, Orlando, Florida, December 2002, pp. 158-162; http://www-eph.int-evry.fr/~horain.

[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, May 2003, pp. 27-31; http://www-eph.int-evry.fr/~soares/WB.

[4] 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 (WMVC)*, Orlando, Florida, Dec. 2002, pp. 118-124; http://www.cs.sunysb.edu/~samaras.

[5] 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.

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

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

[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.

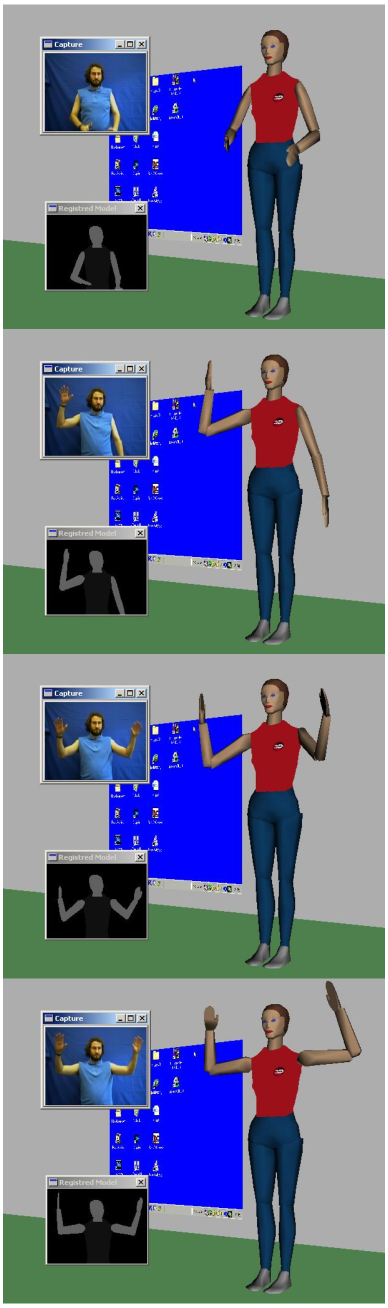


Figure 5. Animation of an avatar from gestures acquired in real time