Enhancing the Perception of Collaborative Actions with Virtual Gestures

Patrick Horain[1]   José Marques Soares[2]   Piyush Kumar Rai[3]   André Bideau[4]

GET / INT, Évry, France

ABSTRACT

This paper proposes using virtual reality to enhance the perception of actions by distant users on a shared application. Here, distance may refer either to space (*e.g.* in a remote synchronous collaboration) or time (*e.g.* during playback of recorded actions). Our approach consists in immersing the application in a virtual inhabited 3D space and mimicking user actions by animating avatars whose motion can be either generated from user actions on the shared application or from motion capture by a computer vision system we briefly describe. We illustrate this approach with two applications, one for remote collaboration on a shared application and the other to playback recorded sequences of user actions. We suggest this could be a low cost enhancement for telepresence.

CR Categories: I.3.7 Three-Dimensional Graphics and Realism; I.3.2 Graphics Systems;. I.2.10 Vision and Scene Understanding.

Keywords: Avatars, animation, collaborative virtual environments, application sharing, telepresence.

1   INTRODUCTION

Classical computer supported collaborative work environments use text, audio and video for remote interaction. However, these techniques suffer from a fragmented perception of users and the objects they share, and from a limited perception of actions by other users, including *e.g.* pointing gestures.

In this paper, we propose to enhance the perception of "who's doing what" in a collaborative session. We propose object-focused interaction based on non-verbal gesture communication as an approach to support interaction on objects of interest and techniques to improve the perception of user actions..

We demonstrate the applicability of our approach with two applications. The first one is based on remote application sharing using animated avatars acting on the application, so bridging space distance. The other one is based on playing back a recorded sequence of users' actions on an electronic document, using animated avatars in 3D inhabited worlds, so bridging time distance.

We propose to augment collaboration with gesture communication by virtual reality. Gestures help focusing attention

_____

¹ E-mail: Patrick.Horain@int-evry.fr.
² Now with Centro Federal de Educação Tecnológica do Ceará (CEFET-CE). E-mail: marques@cefetce.br.
³ Currently a PhD student at the Michigan State University. E-mail: piyush@cse.msu.edu.
⁴ E-mail: Andre.Bideau@int-evry.fr.

on objects of interest. Seeing somebody manipulating an object, rather than just observing the result, may help perceiving that action. Finally, remotely reproducing user gestures allows non-verbal communication [1].

We describe a virtual environment for immersing a shared application in an inhabited world. Distant users are represented with animated avatars acting on the application or by rendering 3D user-motion captured by a computer vision system we briefly describe. We also discuss some usages.

2   APPLICATION SHARING WITH AVATAR ANIMATION

Multipoint videoconferencing-based collaborative applications allow gesture communication but using one video stream per participant requires broadband networking. Such systems poorly support immersion because each user appears in a separate window and it is difficult to know who is acting [2].

Group perception can be improved by immersing the shared application into a virtual multi-user 3D world. In NetICE [3], each user is represented with a humanoid avatar standing by the application mapped on a board in the virtual world. Unfortunately, users' actions on the application are not associated with avatar animation. A further limitation is that the application board is usually poorly readable in the 3D world, so NetICE requires switching between the 3D interface and the 2D application view.

We propose to enhance perception of who is doing what by animating avatars in the collaborative virtual environment to mimic user actions on the application space. The active avatar stands in front of the board and follows with its hand the position of events associated to the user's actions in the application window. In the usual case of single user applications, only one participant can be active so inactive participant just stand waiting on the side. When a non-active user requests to interact, the hand of his avatar is raised to demonstrate his intention. When the active user returns, the avatar of the next active user is moved in front of the board, so participants can directly see who is doing what with the shared application (Figure 1).
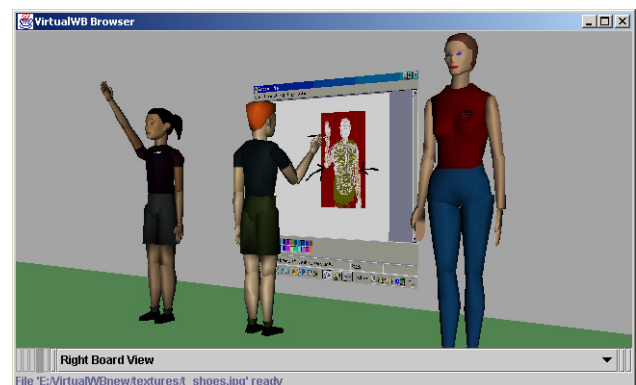

Figure 1. The avatar of the active user shows actions on the virtual board while an inactive user is requesting access and a third user is just observing.

Furthermore, perceiving other participants should be preserved even while acting on the application window. Thus, we have developed a hybrid interface with two parts: an application space that is a high quality view of the application that can be directly manipulated, and an immersive inhabited space that is the virtual meeting place gathering participants and the application they share (Figure 2).
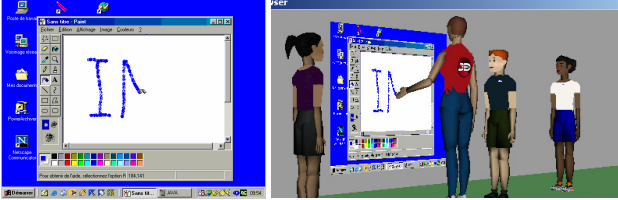

Figure 2. Collaboration in a virtual inhabited world: user actions on the shared application are displayed with computed avatar animation.

We have developed a prototype based on open source VNC [4] for shared remote access to an application. The immersive inhabited space is a VRML model with H-ANIM [5] avatars displayed in custom clients based on the Xj3D free software [6]. A custom events server collects user actions on the shared application and forwards them to remote clients, which in turn locally animate avatars. The IKAN inverse kinematics library [7] [8] is used to compute arm movements of the active avatars so that its hand on the virtual board follows the mouse track on the application (Figure 3). Implementation details are discussed in our previous paper [9].
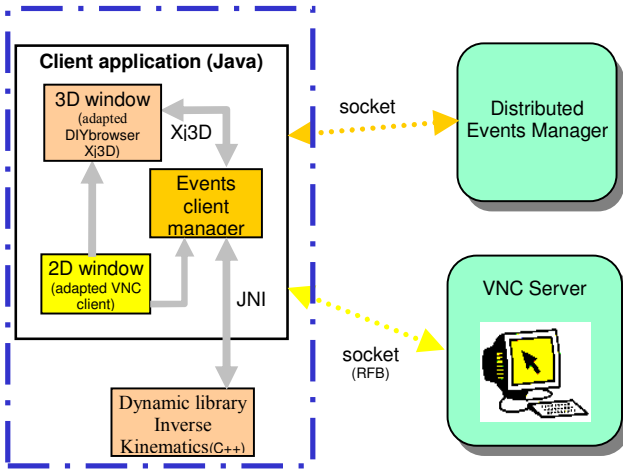

Figure 3. Our collaborative system software architecture.

3   USER-DRIVEN ANIMATION

3.1   Predefined gesture

While active users have their avatar animated from application events, inactive users can also animate their avatar with a set of predefined expressive gesture (*e.g.* welcome, laugh, applause, disagreement, etc) that are stored and transmitted in MPEG-4/BAP format [10].

3.2   Motion capture by computer vision

We have also developed a motion capture by computer vision system to further animate avatars.

Our approach for motion capture works with a single camera, without markers and without *a priori* restriction to a set of gesture [11] [12]. It consists in registering an articulated 3D model of the human upper body onto a video sequence.

This system is described hereafter. The final subsection shows how gesture acquired with a webcam can be remotely reproduced by animating virtual actors in collaborative virtual environment.

3.2.1   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. For each color class, a hue histogram is initially learnt from sample video image parts, 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 [13]. Figure 4 shows a probability image for the skin color class (b) corresponding to image (a).

Image segmentation is obtained by setting each pixel to the class with highest probability. A morphological opening is used to reduce noise.
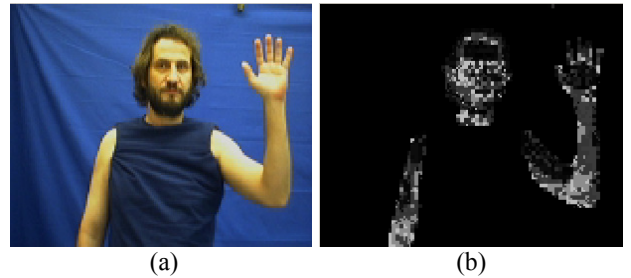

Figure 4. Video image (a) and skin color probability (b).

3.2.2   Projection of the 3D model

The 3D-model has 23 degrees of freedom. Biomechanical limits constrain the registered postures to be morphologically reachable. Each segment of the model is associated with a color class. Computing the projection of the 3D colored model in the image plane is accelerated with the graphic card. 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.

3.2.3   Comparing the model projection with the segmented image

Optimal registration is searched by minimizing the mismatch between the segmented video image (Figure 5a) and the image projection of the 3D-model (Figure 5b). Model candidate postures are evaluated against the segmented video images by comparing their color features. For that purpose, we minimize a non-overlapping ratio:

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

where q is the vector of joints parameters describing a candidate posture, A_c is the set of pixels in the cth color class in the segmented video image, $B_c(q)$ is the projection of the model segments with the cth 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 $B_c(q)$ are computed by comparing the pixels in the segmented video image
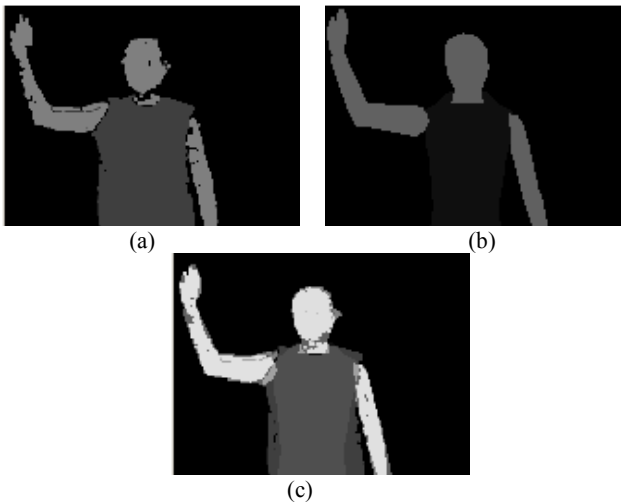
(a)

(b)

(c)

Figure 5. (a) Segmented video image; (b) projected model image; (c) their superposition (sum).

and the 3D model projection images for all the color classes c. We achieve this 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 color classes and background, pixels labels can be stored in the 4 most significant (respectively least significant) bits of a byte. We combine the segmented and projected images by adding them in a single superposition image (Figure 5c). Adding these 2 images generates a superposition image (Figure 2c) where each pixel value indicates the intersection $A_x \cap B_y(q)$ it belongs, where x and y are a color class or the background. An histogram operation gives the number of pixel in each $A_x \cap B_y(q)$. $A_c \cup B_c(q)$ in expression (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 with their 4-bits hexadecimal labels, the number of pixels of $A_c \cup B_c(q)$ is the sum of the histogram bins with 8-bits hexadecimal labels cx and yc where x and y can be any color class or the background, $y \neq c$.

For acceleration purpose, both the 3D model projection and the superposition image are computed with the graphical processing unit in consumer PCs through with plain OpenGL programming.

3.2.4 Detecting 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. To reduce the execution time, we detect the moving model parts and adjust only their parameters, so decreasing the parameters space dimension.

We define groups of model segments (bust, right arm, left arm, and head) and we use different labels for the colors of these groups. Groups whose intersection cardinals have significantly changed from one image to the next are considered to be in movement. For example, Figure 6 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 3D model parts hierarchy.

3.2.5 Optimal registration

Optimizing the registration is computationally intensive because our evaluation function cannot be analytically derived. Thus, gradient descent algorithms [14] [15] cannot be used. For this reason, we use a downhill simplex algorithm [16] that requires

Figure 6. Superposition of the 3D model registered at time t with the segmented video images at t and t+1.

evaluation of a large number of candidate postures. In counterpart, this evaluation function can be implemented very efficiently using accelerated graphic cards of a standard PC and also using libraries optimized for the SIMD instructions supported by modern processors.

3.3 Integrating motion capture into the 3D environment

The 3D gesture acquisition system outputs the posture parameters acquired for each image in the MPEG-4/BAP [10] format. Then, body motion is rendered by remote client virtual world by animating the associated avatar. This allows gesture-based communication in the shared virtual environment without the limitation of predefined animations (Figure 7).

4 APPLICATION TO PLAYBACK RECORDED SEQUENCES

Distance to events may as well refer to time rather than space, *e.g.* when reviewing a conference or a lecture from recorded slides, annotations and voice. Electronic documents with animations and synchronized audio can be recorded with SVG or XML [17] for later review. We propose to enhance the perception of reviving such an event by doing a virtual playback. The electronic documents being reviewed can be immersed in a virtual inhabited space (*e.g.* a virtual classroom) and the interactions made by different individuals, on such an electronic document (stored in SVG format, for example) can be mimicked by animated 3D humanoids. The objective is to reproduce all such past interactions on the electronic document by doing a "playback", using animated avatars.

The SVG format supports documents as time-stamped images, text and graphics including annotations as paths with attributes that allow encoding some author identity. Annotations, here, refer to any type of graphical overlay that appears on a document, like scribbling on some section of a lecture slide, encircling or "tick-marking" some keywords or diagrams on the lecture slides, as is done on a paper based document using a pen or pencil. In our case, the electronic documents (in SVG format) are classroom lecture slides and the equivalent annotations are encoded as "path" elements according to SVG specification. The path element used for defining annotations consists of coordinates of all the points an annotation is made of, which include starting, ending and all intermediate coordinates (as defined by SVG specifications).

The application, we have developed, will enable playing back the entire content (stored and encoded in SVG format) of a classroom lecture in an immersive 3D environment (developed in VRML / X3D) inhabited by animated 3D avatars at any later point of time. These avatars will be acting as agents interacting with an object: a virtual whiteboard, which would correspond to an actual whiteboard in a real classroom setting. The method does not require high-bandwidth requirement as opposed to video or other storage and replaying mechanisms, because of minimal storage (due to small size of SVG files) required. All the events and contents of the classroom lecture are exactly reproduced. In short,
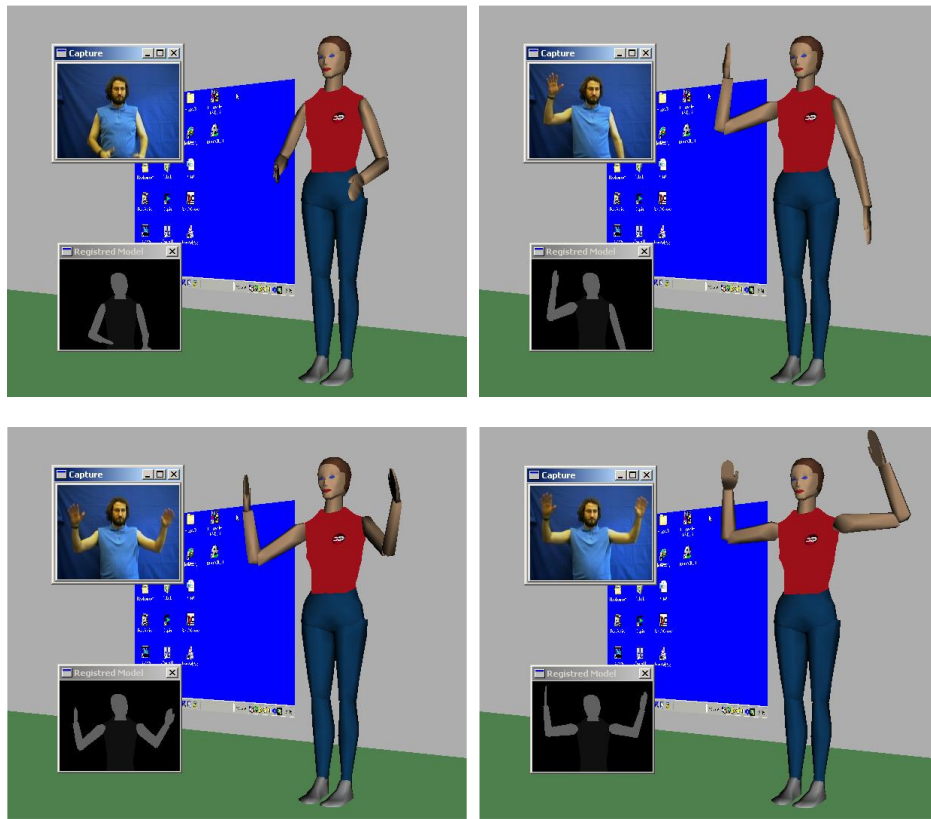
Figure 7. Animation of an avatar from gestures acquired in real time.

the application reproduces "all what happened in the classroom lecture".

4.1 Storage format for the documents

The SVG format enables storing the lecture slides with time-stamps for each slide and also for each annotation which appears on that slide. The following SVG snippet shows two lecture slides having one annotation on each. Annotations on each slide can be categorized as either Instructor's Notes, or Student's Notes. Instructor's notes on a slide are annotations marked by the instructor. Similarly, Student's notes are annotations marked by a student. A slide can have annotations by instructors and student, both.

For the sample SVG document shown in this example below, the first slide appears 3.221 seconds after the document has been loaded. The first annotation on this slide starts getting drawn after 8.998 seconds. The time at which an annotation drawing is finished on the document is also specified. There can be more annotations on the same slide. Similarly, the time at which the slide itself ceases to show is also specified. After that, there can be new slides with different annotations on them. The slides and the annotations can appear on the document in any order determined only by the timing of appearance (and not by their order in the SVG document). Also, same slide can appear multiple times with different annotations on it while the content is played back (for example, once under instructor's notes, and again under student's notes).

```
<g id='instructor_notes'>

//SLIDE: 1
<g id='slide_0' display='none'>
<set attributeName='display' to='inline' begin='3.221'/>
<image xlink:href='bovary.jpg' width='640' height='480'
style='font-size:12;' id='image1' />
<g id='annot_1_1' display='none'>
<set attributeName='display' to='inline' begin='8.998'/>
<path id="curve2" d="M 64 217 L 63 217 L 63 218 L
62 226 L 61 239 L 60 257 L 60 277 L 60 297 L 64 310
L 67 317 L 70 320 L 71 320 L 73 320 L 79 317 L 87
310 L 96 303 L 107 293 L 115 284 L 122 275 L 127
265 L 132 257 L 134 251 L 134 249 L 135 250 L 135
256 L 135 265 L 137 276 L 140 289L 143 304 L 147
316 L 151 327 L 156 333 L 158 335 L 159 335 L 160
335 L 162 335 L 166 331 L 169 327 L 174 321 L 179
313 L 184 305 L 187 297 L 190 290 L 192 284 L 193
279 L 194 275 L 194 273 L 194 271 L 195 271L 195
269 L 195 268 L 197 268 L 204 274 L 217 284 L 240
297 L 269 312 L 309 330 L 350 344 L 387 353 L 418
361 L 441 366 L 450 367" style="fill:none;fill-
rule:evenodd;stroke:#00ffff;stroke-width:3" />
<set  xlink:href='#annot_1_1'  attributeName='display'
to='none' begin='16.974'/>
</g>
<set attributeName='display' to='none' begin='18.239'/>
</g>
```

```
// SLIDE: 2
<g id='slide_1' display='none'>
<set attributeName='display' to='inline' begin='18.239'/>
<image xlink:href='index.jpg' width='640' height='480'
style='font-size:12;' id='image2' />
<g id='annot_2_1' display='none'>
<set attributeName='display' to='inline' begin='19.750'/>
 <path       id='annotation_2_1'       style='fill:none;fill-
rule:evenodd;stroke:yellow;stroke-width:3;'
 d='M 287 48 L 287 49 L 287 50 L 285 55 L 282 63 L
276 74 L 269 88 L 259 103 L 248 121 L 234 136 L 217
151 L 199 164 L 176 175 L 152 180 L 131 180 L 112
176 L 103 167 L 100 157 L 99 146 L 99 130 L 103 114
L 111 94 L 124 75 L 135 57 L 148 43 L 158 38 L 162
35 L 163 35 L 168 39 L 177 53 L 193 77 L 220 117 L
254 168 L 286 212 L 306 239 L 319 257 L 328 266 L
330 270 L 331 271'/>
<set  xlink:href='#annot_2_1'  attributeName='display'
to='none' begin='21.974'/>
</g>
```

4.2   2D + 3D Application Immersion

A 2D application parses the document based on the appearance timing of each slide and each annotation on this slide and shows the slides and the annotations at pre-specified timings in its browser. We have used the open source Apache Batik toolkit [18] for core SVG browsing. Each annotation is shown in the 2D application as appearing in a progressive manner so as to mimic the actual drawing by the entity which drew it in real world. The 2D application sends each event-coordinate and corresponding timings to another 3D application having pre-loaded avatars, one corresponding to each acting entity in the real world. These avatars are defined according to H-Anim specification. Event-coordinates refer to the current drawing coordinates of the annotation in the 2D application browser and timings refer to the time elapsed from the beginning of drawing the annotation. The 2D application is integrated and immersed in a 3D world using X3D/VRML. The window of 2D application, showing slides and annotations appearing on it, is mapped onto a virtual whiteboard in this 3D world. The event-coordinates and corresponding timings are then used to animate the avatars using inverse kinematics and to reproduce the exact behavior of the real entities (instructor and students) on the electronic document.

5   CONCLUSION AND PERSPECTIVES

We have described an approach to enhance collaborative work or playback presentations by displaying user actions on a shared application as avatar actions in a 3D virtual world. It achieves gesture-based communication with little bandwidth or storage by animating avatars on the fly from application events. The system runs on consumer PCs. Some demonstration videos are available at http://www-eph.int-evry.fr/~horain/MarquesSoares.

The collaborative version augmented with voice over IP was briefly tested at INT for remote learning. A lecture was given jointly by 2 teachers in 2 classrooms, each teacher with a group of students. Rather than using 2 cameras and a broadband video communication, the slides and virtual world were shared through our system and video projected in both rooms, as showed in Figure 8. A large majority of students stated they would volunteer to attend again lectures with that technology [12].
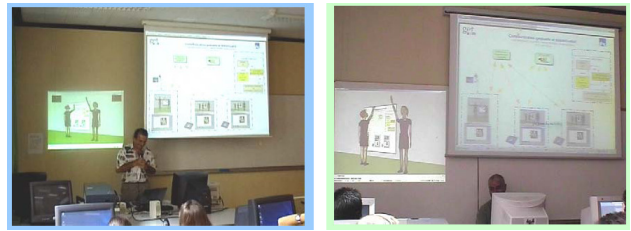

Figure 8. Lecture given jointly by 2 teachers in 2 classrooms.

More than just an artifact extra communication channel, the collaboration system can also be the basis for remote perception of the real world. Consider video projecting the shared application window and capturing user's actions on the projection board with a wireless pen system such as a *mimio* [19]. The virtual view appears to be close to the real scene (Figure 9). This could be a basis for remote presence with low cost equipment for enhanced collaboration around a physical board.
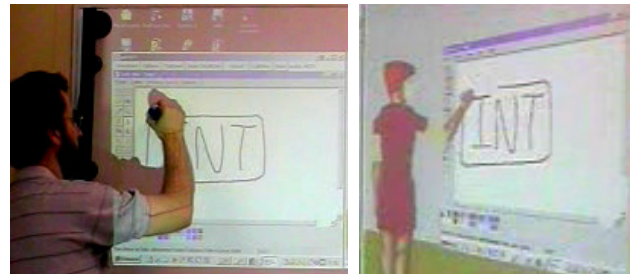

Figure 9. Remote virtual animation from an augmented board.

The shared applications we used were 2D so far, but we are considering an extension for collaborative 3D visualization. Also, making avatars look like the users they represent would allow a straightforward association in a more user-friendly system. Finally, off-the-screen gestures may also be valuable for communication, so we are developing a motion capture interface by real time single-view marker-free computer vision to remotely control avatars [12].

REFERENCE

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

[2]   W. H. Leung and T. Chen. Creating a Multiuser 3-D Virtual Environment. *IEEE Signal Processing Magazine*, vol. 18, n° 3, May 2001, pages 9–16.

[3]   Advanced Multimedia Processing Lab. *Project NetICE*. http://amp.ece.cmu.edu/projects/NetICE.

[4]   RealVNC. *Virtual Network Computing*. http://www.realvnc.com.

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

[6]   Web3D Consortium. *The Xj3D project*. http://www.xj3d.org.

[7]   Center for Human Modeling and Simulation. *IKAN: Inverse Kinematics using ANalytical Methods*. http://hms.upenn.edu/software/ik.

[8]   D. Tolani, A. Goswami and N. Badler. Real-time inverse kinematics techniques for anthropomorphic limbs, *Graphical Models*, Vol. 62, No. 5, 2000.

[9]   J. Marques Soares, P. Horain and A. Bideau. Sharing and immersing applications in a 3D virtual inhabited world", *Proc. VRIC 2003*, Laval, France, May 2003, pp. 27-31.

[10] A. E. Walsh and M. Bourges-Sévenier. *MPEG-4 Jump-Start*. Prentice Hall, Upper Saddle River (2002).

[11] P. Horain and M. Bomb. 3D Model Based Gesture Acquisition Using a Single Camera. *Proc. of IEEE Workshop on Applications of Computer Vision (WACV 2002)*, Orlando, 2002, pp. 158-162. http://www-eph.int-evry.fr/~horain.

[12] J. Marques Soares. *Contribution à la communication gestuelle dans les environnements virtuels collaboratifs*. Ph. D. Thesis n° 2004INT0002, INT, July 2004. http://www-eph.int-evry.fr/~horain/MarquesSoares.

[13] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2nd Quarter 1998. ftp://download.intel.com/technology/itj/q21998/pdf/camshift.pdf.

[14] S. Lu, G. Huang, D. Samaras and D. Metaxas. Model-based Integration of Visual Cues for Hand Tracking. *Proc. of IEEE workshop on Motion and Video Computing (WMVC)*, Orlando, 2002, pp. 118-124.

[15] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 2003. Vol. 22, n° 6, pp. 371-293.

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

[17] J.-C. Moissinac. Traitement automatisé de conférences. *Proc. CORESA 2004*, Lille, mai 2004. http://www-rech.enic.fr/coresa04/programme.html#indexation.

[18] Apache. *Batik SVG Toolkit*. http://xml.apache.org/batik.

[19] Virtual Ink. *mimio: Interactive Whiteboards*, http://mimio.com.