Perceiving and rendering users in a 3D interaction

Patrick Horain, José Marques Soares, Dianle Zhou, Zhenbo Li, David Antonio Gomez Jauregui, Yannick Allusse

Perceiving and rendering users in a 3D interaction

Patrick Horain, José Marques Soares, Dianle Zhou, Zhenbo Li,
David Antonio Gómez Jáuregui and Yannick Allusse

Institut Telecom / Telecom SudParis, 9 rue Charles Fourier, 91011 Evry Cedex, France
Patrick.Horain@Telecom-SudParis.eu

Abstract. In a computer supported distant collaboration, communication between users can be enhanced with a visual channel. Plain videos of individual users unfortunately fail to render their joint actions on the objects they share, which limits their mutual perception. Remote interaction can be enhanced by immersing user representations (avatars) and the shared objects in a networked 3D virtual environment, so user actions are rendered by avatar mimicry. Communication gesture (not actions) are captured by real-time computer vision and rendered. We have developed a system based on a single-webcam for body and face 3D motion capture. We have used a library of communication gestures to learn statistical gesture models and used them as prior constraints for monocular motion capture, so improving tracking ambiguous poses and rendering some motion details. We have developed an open source library for real-time image analysis and computer vision that supports acceleration by consumer graphical processing units (GPUs). Finally, users are rendered with low-bandwidth avatar animation, thus opening the path to low-cost remote virtual presence at home.

Keywords: 3D interaction networked virtual environment, motion capture, mocap, face tracking, computer vision, GPU acceleration, telepresence.

1 Introduction

Computer supported remote collaborative work environments usually allow limited perception of actions by other users. While multipoint videoconference allows perceiving remote users at the cost of some network bandwidth, it poorly supports immersion because each user appears in a separate window and it is difficult to know who is acting on the shared objects [1].

We aim at enhancing the perception of "who is doing what" during computer supported collaborative work remote session. Our approach is to gather users and the graphical objects they share in a multi-user 3D virtual space and to augment collaboration by virtual rendering user actions and motion. Gestures help focusing attention on objects of interest. Seeing somebody manipulating an object, rather than just observing the action output, will help perceiving that action. Furthermore, distant restitution of user gestures in a collaborative virtual environment allows non-verbal communication [1].

We first describe an environment to immerse a shared 2D application in an inhabited virtual world, so virtually gathering users and the shared objects in a single place. Distant users are represented with animated avatars acting on the application.

When a user is not acting on the application, his avatars should still be kept alive and rendering user motion. We have completed the previous environment with a perceptive interface to capture user face and body 3D motion in real-time using just a webcam.

Finally, we address the issue of real-time image processing and analysis with consumer graphical processing units (GPUs) and describe *GpuCV*, our open source library for GPU-accelerated computer vision.

2 Remote virtual 3D collaboration

Group perception can be improved by immersing the shared application into a virtual multi-user 3D world. In NetICE [2], 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 aim at enhancing perception of "who is doing what" by animating avatars in the collaborative virtual environment so they mimic user actions on the application. The avatar of the active user stands in front of the board. It is animated with inverse kinematics so that his hand follows the event position of the user's actions in the application. In the case of single user applications, avatars non-active users willing to interact raise a hand to show their request.

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 (Fig. 1). Implementation

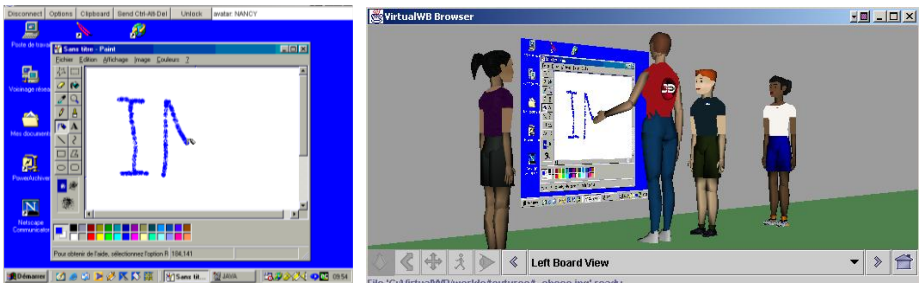

Fig. 1. Remote collaboration in a virtual inhabited world: a shared application and representations of users are gathered in a single place. User actions on the application are rendered by avatar animation. Demonstration videos can be found at http://www-public.it-sudparis.eu/~horain/MarquesSoares/WB.

details are discussed in our previous paper [3]. It allows to render user actions with little bandwidth.

The collaborative version augmented with voice over IP was briefly tested 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. A large majority of students stated they would volunteer to attend again lectures with that technology [4].

More than just an artifact extra communication channel, this system can also be a low cost approach for remote perception of a real scene. Consider video projecting the shared application window and capturing user's actions on the projection board with a wireless pen system [5]. The virtual rendering can be close to the real scene (Fig. 2). This allows presence with low cost equipment and a low bandwidth communication channel for enhanced remote collaboration around a physical board.

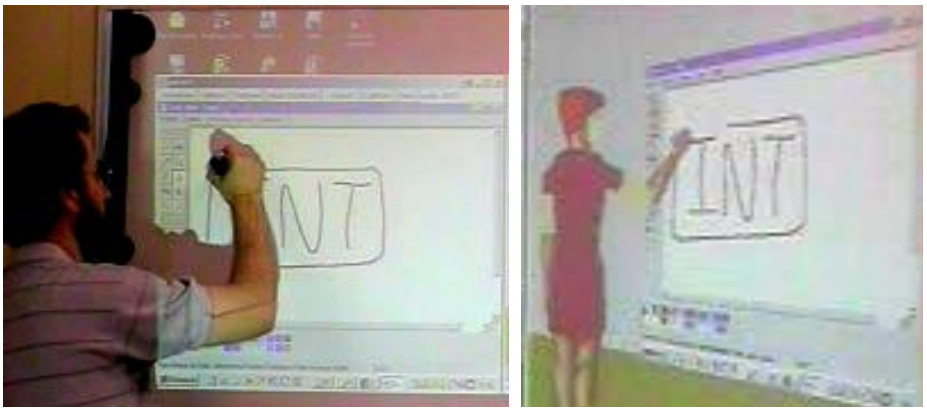

Fig. 2. Remotely rendering user actions from an augmented board. A demonstration video is available from http://www-public.it-sudparis.eu/~horain/MarquesSoares/WB.

3 Real-time body motion capture with a webcam

When the user is not active on the shared application, his gesturing still has to be captured in some way and rendered virtually. Marker-less human motion capture by computer vision allow inexpensive, non-obtrusive body tracking [6]. Research in this field has been motivated by many target applications: human-computer interfaces, animation, interaction with virtual environments, video surveillance, games, etc [7].

We have developed a real-time webcam-based system for 3D motion capture. It works by registering an articulated 3D model of the human body on a video stream (Fig. 3).

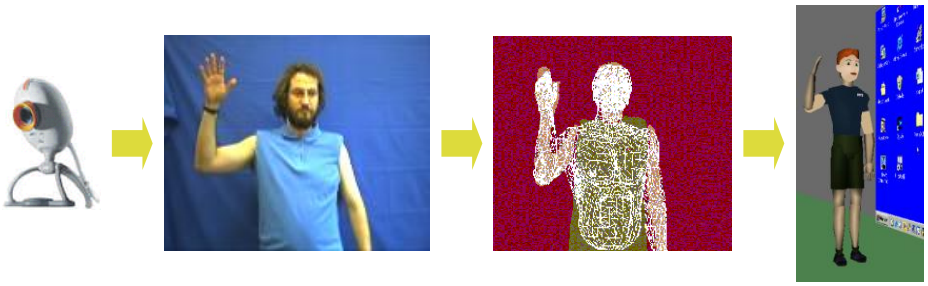

Fig. 3. Real time motion capture by computational vision and virtual rendering.

Our 3D human model has 3 global position parameters and 20 joint angles of the upper-body (bust, arms, forearms, hands, neck and head), so a body-pose is represented by a vector of 23 parameters. For each input image we search for the model pose that best matches the image. Image features (colour regions, edges) are extracted and matched with model features (coloured limbs, occluding edges) projected in the candidate pose. For each captured image, optimal registration is searched with respect to the pose parameters by first iteratively maximizing the colour region overlap and refining registration by minimizing the distance between the image edges and the projected occluding edges of the model (Fig. 4) [8] [9].

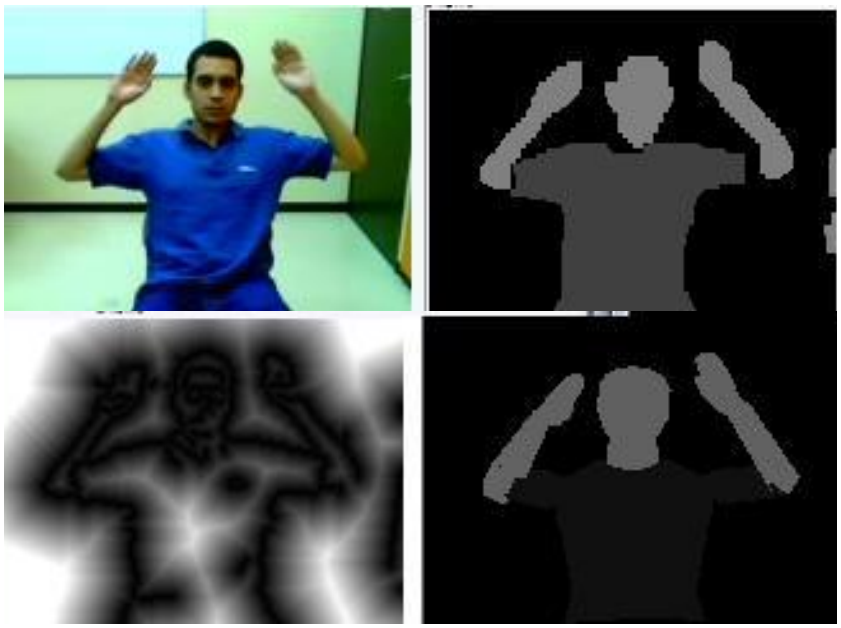

Fig. 4. The captured image, the color-segmented image, map of distance to image edges in the foreground mask and finally the projection of the 3D human body model in the optimal matching pose.

Joint angles are then output in real-time over the network as low bandwidth MPEG-4 body animation parameters (BAPs). They are rendered remotely the captured motion by animating the user avatar in the virtual space (Fig. 5).

Fig. 5. Real-time motion capture using a single camera for each user and virtual rendering. Demonstration videos are available from http://MyBlog3D.com.

4 Real-time face tracking and 3D avatar animation

Face-to-face communication includes visually perceiving the face of engaged people. When interacting in a virtual environment, face perception should be supported trough that environment which implies detecting human faces, tracking their motion, and rendering animated faces. This is a challenge because face appearance varies with the head pose (including out-of-plane head rotations), facial expressions, lighting conditions, occlusions, or combinations of them.

Feature-based face tracking proceeds by matching local features. It can cope with illumination changes and affine deformations of the image. Instead, active appearance models work by globally registering a face model onto images. They usually require an extensive initial learning of the appearance and its variations from example images of the observed user. Appearance variations are learnt with respect to model parameters, so parameter variations can then be iteratively estimated to minimize the residual appearance difference between the reference input image and the model-

generated appearance. Face tracking then consists in optimally registering the face model that was learnt onto input images using a gradient descent. The active appearance approach has been found be more precise than the feature-based approach [10].

We have used an active model approach for real-time 3D head pose and face expression tracking with a webcam (Fig. 6). We have used a statistical approach to dynamically fit the appearance model to unknown users by estimating the gradient and update matrices from the face texture. We have implemented this algorithm for tracking of the head position (6 parameters) as well as face motion (6 parameters in real-time) and experimentally demonstrated its robustness when tracking multiple or unknown users' faces [11].

Fig. 6. Real-time 3D face tracking. Demonstration videos are available from http://MyBlog3D.com**.**

Then the output face parameters are encoded in MPEG-4 face animation parameters (FAPs) so they can be sent to a remote player that will animate the face of the user avatar (Fig. 7).

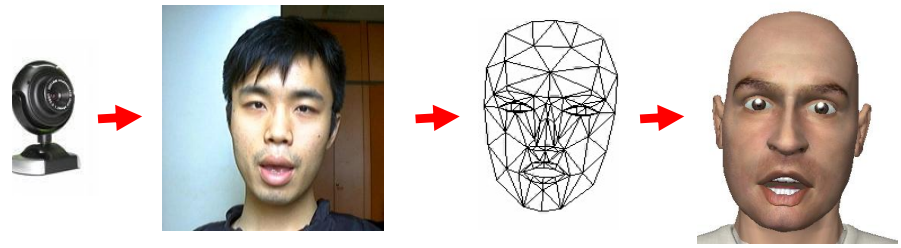

Fig. 7. Face tracking by registering the Ahlberg's CANDIDE-3 face model [12] and rendering with the GRETA player by Pelachaud *et al.* [13]. Demonstration videos are available from http://MyBlog3D.com.

5 Statistical gesture modeling

Monocular video based markerless 3D motion capture has the intrinsic limitation of poor precision and robustness. . This is a difficult problem because of the ambiguities resulting of the lack of depth information, partial occlusion of human body parts, high number of degrees of freedom, variations in the proportions of the human body and various clothing that may confuse the tracker. We have used statistical gesture models as guides to constrain and disambiguate 3D tracking for a set of human motions.

Although people can perform very large variation of complex motions, their movements can be represented in a low-dimensional latent space, that is a subspace with a lower dimension than the full motion capture data [14] [15]. Because human motion is non-linear, basic dimensionality reduction methods such as principal component analysis (PCA) are inadequate to describe non-linear human motion [6]. Locally Linear Embedding (LLE) and Isomap either do not provide invertible mapping from the low dimensional latent space to the original pose space or do not provide probability distribution over data in latent space, so they are not suitable to build low dimensional gesture models [16]. Gaussian Process Latent Variable Models (GPLVM) [17] and then Gaussian Process Dynamic Models (GPDM) [16] can learn a non-linear mapping between the human motion parameter space and a latent space and so they provide an inverse mapping. They allow describing human motion in a low-dimensional latent space. Recently, latent gesture models have been used as prior constraints to help 3D human motion tracking. Urtasun [18] used GPLVM and GPDM to learn prior models for tracking 3D human walking. She achieved good results even in case of serious occlusion.

Training these models requires relevant gesture databases. Most existing databases as CMU Graphics Lab Motion Capture Database [19] and HumanEva [20] gather data on human actions such as running, jumping and the like, but very few include communicative gestures. Since we aim at visually tracking communicative gesture rather than actions, we learnt gesture models from a set of example conversational gestures with variations [21] that was generated using the Greta expressive conversational agent [13].

We modeled gesture with Gaussian Process Dynamic Models (GPDM) [16], a powerful approach for probabilistically modeling high dimensional time related data through dimension reduction. GPDM can learn probability motion models from small training data sets as a mapping between the full-dimensional data space and a low-dimensional latent space and a dynamical model in the latent space. Fig. 8 shows two conversational gestures from the gesture library and their corresponding gesture trajectories in 3D latent space. Each circle on the trajectories corresponds to a body pose.

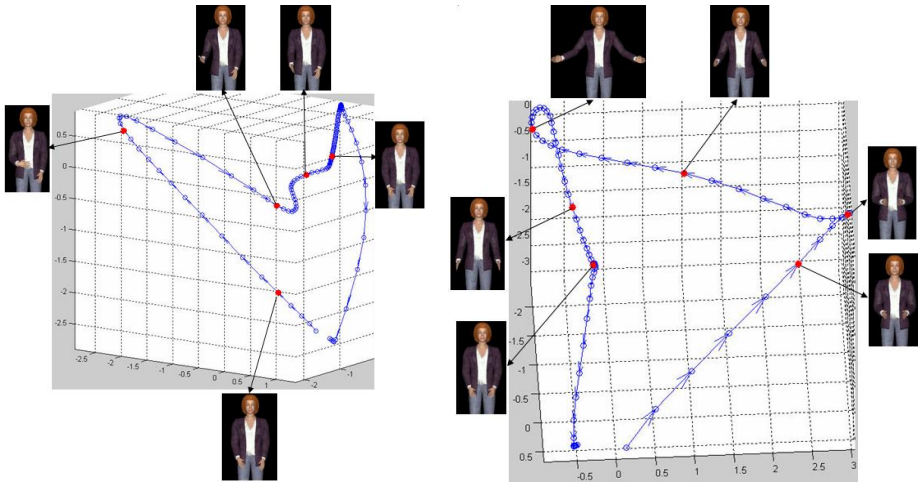

Fig. 8. Conversational gestures described in a 3D latent space. Each circle on the trajectories represents a body pose.

Those gesture models were then used as constraints while tracking specific gestures. Each candidate pose is projected into the latent gesture trajectory, so enforcing the captured motion regularity.

For each iteration, the motion tracker outputs a candidate pose to be constrained with the gesture model. That pose is projected from the full motion parameter space to the 3D latent model space and then replaced with the closest point on the latent motion trajectory. Since GPDM mapping is continuous, poses that are close in the full motion parameter space remain close in the latent space. The output constrained pose at each time step is the pose reconstructed into the full motion space. The point used for pose reconstruction is the closest point from the projected point on the model trajectory in the latent space. The resulting pose is then used as the initial pose for tracking at the next image (Fig. 9).

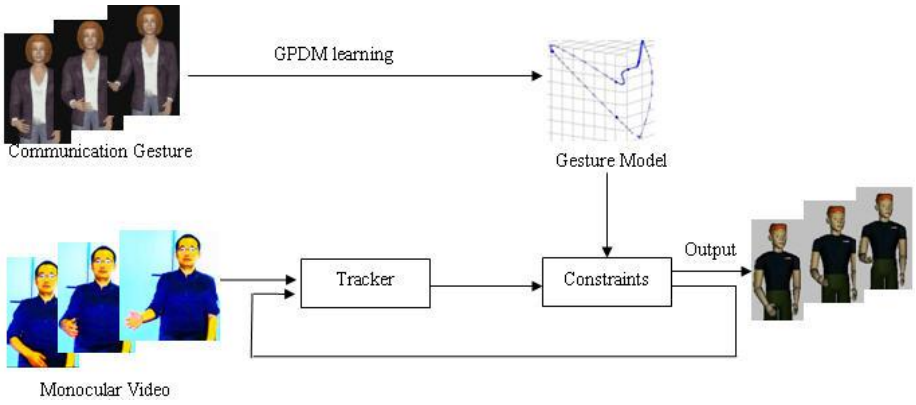

Fig. 9. Gesture model working as constraints in tracking

Because the output poses lay on the gesture model internal trajectory, most of the monocular vision ambiguities can be solved. Impossible poses will not happen in the gesture model space, so heuristic biomechanical constraints, which otherwise must be used for pruning the pose space, can be replaced with the gesture model that constrain the output poses to be on the learnt motion trajectory (Fig. 10). Another benefit of this approach is that the poses reconstructed in the latent space include motion details that cannot be captured from the input video sequence (such as hand shapes) (Fig. 11) [21].

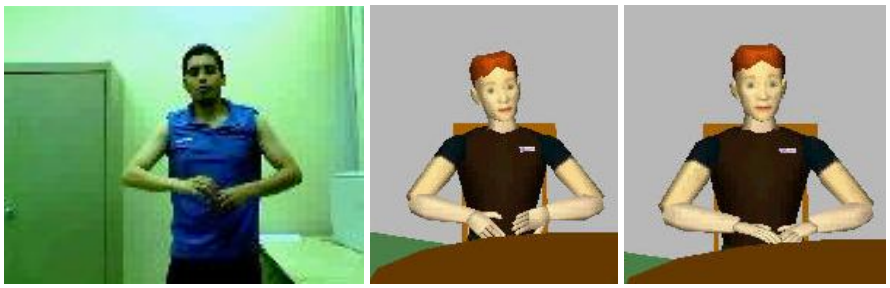

Fig. 10. Motion tracking with biomechanical constraints *vs.* gesture model. **Left:** An input image where the left hand orientation can hardly be distinguished. **Middle:** Motion capture result with heuristic biomechanical constraints: the unlikely pose of the left hand is biomechanically possible, so it is accepted. **Right:** Tracking with gesture model instead of biomechanical constraints: the gesture model avoids the awkward pose.

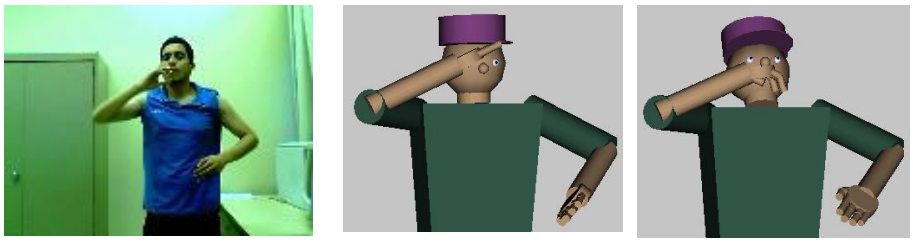

Fig. 11. Augmenting motion capture with gesture models. **Left:** Input image. **Middle:** Tracking without gesture model: the hand shape is not captured to meet real-time computation constraints. **Right:** The gesture model augments the motion capture with fingers movements.

6 GPU acceleration for image processing

These real-time computer vision algorithms require more computing power than even high end CPUs can offer. Registering 3D models can be accelerated with graphical processing units (GPUs) that have made their way to consumers PCs through video games and multimedia, and are now coming to cell phones.

We harnessed this readily available resource for image processing and analysis. Since 2005, we have developed GpuCV, an open-source GPU-accelerated image processing and Computer Vision library [22] [23]. It is meant to help computer vision scientist porting existing applications or developing new algorithm to GPU, so taking advantage of the high level of parallelism and computing power available from GPUs, without bothering with the complexity of GPUs.

GpuCV programming interface is compatible with the popular OpenCV library [24] for CPU-based applications. It supplies a set of GPU-accelerated OpenCV-like operators that are ready to be used in OpenCV applications as well as a framework for creating custom GPU-accelerated operators based on OpenGL GLSL or NVIDIA CUDA APIs. It can transparently manage hardware capabilities and data synchronization. A GpuCV specific and innovative feature is that it supports multiple implementations for a single image processing routine with on-the-fly benchmarking and switching to the most efficient implementation.

It is available as a free software under the CeCILL-B license from http://picoforge.int-evry.fr/projects/gpucv.

7 Conclusion and perspective

We have described an approach to enhance remote interaction by gathering user representations and the objects they share in a single virtual place, and by rendering users from their action on objects and from their motion perceived with computer vision. These tools are demonstrated to work in real-time with consumer PCs and webcams. This allows low-bandwidth distant gesture-based communication through virtual environments.

This opens a new communication channel between distant people that can be a basis for remote presence and for enhanced interaction and collaboration with low cost equipment.

Such a perceptive interface could even make its way to the mass market. While networked 3D virtual environments such as Second Life [25] generated quite a buzz a few years ago, attention now rather goes to social networks with more conventional interface. Indeed, avatars are meant to be the virtual counterpart of human users, but the current technology only offers difficult and tedious control and animation of those representatives. User perception system we described allows achieving full avatar control in real time with just consumer hardware, at home.

Acknowledgments. We thank Pr. Catherine Pelachaud and her team for providing the GRETA player [13] and the corpus of conversational gesture we used for modeling gesture [21]. Dr. Jörgen Ahlberg provided the CANDIDE-3 face model [12]. José Marques Soares was supported by the Brazilian Government through CAPES/COFECUB project n°266/99-I.

References

1. Guye-Vuillème, A., Capin, T. K., Pandzic, I., Thalman, N., Thalman, D.: Nonverbal Communication Interface for Collaborative Virtual Environments. *Virtual Reality J.*, vol. 4, 49-59 (1999). [doi:10.1007/BF01434994].
2. Leung, W. H., Chen, T.: Creating a Multiuser 3-D Virtual Environment, *IEEE Signal Processing Magazine*, vol. 18, n° 3, 9-16 (2001). [doi:10.1109/79.924884].
3. Marques Soares, J., Horain, P., Bideau, A.: Sharing and immersing applications in a 3D virtual inhabited world. In: *Proc. VRIC 2003*, Laval, France, pp. 27-31 (2003).
4. Marques Soares, J., *Contribution à la communication gestuelle dans les environnements virtuels collaboratifs*, Ph.D. Thesis n° 2004INT0002, INT (2004). http://www-public.it-sudparis.eu/~horain/MarquesSoares.
5. mimio Interactive, http://mimio.com.
6. Poppe, R.W.: Vision-based human motion analysis: An Overview. *Computer Vision and Image Understanding*, Vol. 108, Issue 1-2, 4–18 (2007). [doi:10.1016/j.cviu.2006.10.016]
7. Moeslund, T., Hilton, A., Kruger, V., A survey of advances in vision-based human motion capture and analysis, *Computer vision and image understanding*, Vol. 104, Issues 2-3, 90-126 (2006) [doi:10.1016/j.cviu.2006.08.002].
8. Horain, P., Bomb, M.: 3D Model Based Gesture Acquisition Using a Single Camera. In: *Proceedings of IEEE Workshop on Applications of Computer Vision (WACV 2002)*, Orlando, Florida, pp. 158-162 (2002) [doi:10.1109/ACV.2002.1182175].
9. Gómez Jáuregui, D. A., Horain, P.: Region-based *vs.* edge-based registration for 3D motion capture by real time monoscopic vision. In: A. Gagalowicz and W. Philips (Eds.) *Computer Vision/Computer Graphics Collaboration Techniques – Proceedings of MIRAGE 2009*, Rocquencourt, France. LNCS, vol. 5496, pp. 344–355. Springer Berlin / Heidelberg (2009). [doi:10.1007/978-3-642-01811-4_31].
10. Cootes, T., Edwards, G., Taylor, C.: Active appearance models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (6), 681–685 (2001). [doi:10.1109/34.927467].

11. Zhou, D., Horain, P.: Robust 3D Face Tracking on Multiple Users with Dynamical Active Models. In: Huet, B.; Smeaton, A.F.; Mayer-Patel, K.; Avrithis, Y. (Eds.): *Advances in Multimedia Modeling – Proceedings of the 15th International Multimedia Modeling Conference (MMM2009)*, Sophia Antipolis, France. LNCS, vol. 5371, pp. 74-84, Springer Berlin / Heidelberg (2009). [doi:10.1007/978-3-540-92892-8_9].
12. Ahlberg, J.: *CANDIDE - a parameterized face*, http://www.bk.isy.liu.se/candide.
13. Pelachaud, C. *et al.*: *GRETA: Embodied Conversational Agent*, http://www.tsi.enst.fr/~pelachau/Greta.
14. Elgammal, A. M., Lee, C.-S.: Inferring 3D body pose from silhouettes using activity manifold learning. In: *Conference on Computer Vision and Pattern Recognition (CVPR'04)*, vol. 2, pp. 681–688 (2004). [doi:10.1109/CVPR.2004.132].
15. Grochow, K., Martin, S. L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. *ACM Transactions on Graphics* 23 (3), 522–531 (2004). [doi:10.1145/1015706.1015755].
16. Wang, J. M., Fleet, D. J., Hertzmann, A.: Gaussian Process Dynamical Models for Human Motion. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2), 283-298 (2008). [doi:10.1109/TPAMI.2007.1167].
17. Lawrence, N. D.: Gaussian process latent variable models for visualisation of high dimensional data. In: Thrun, S., Saul, L., Schölkopf, B. (eds): *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, pp. 329–336 (2004).
18. Urtasun, R., Fleet, D. J., Fua, P.: 3D people tracking with Gaussian process dynamical models. In: *Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, New York, NY, pp. 238–245 (2006).
19. Carnegie Mellon University, *CMU Graphics Lab Motion Capture Database*, http://mocap.cs.cmu.edu.
20. Brown University, *HumanEva*, http://vision.cs.brown.edu/humaneva.
21. Li, Z., Horain, P., Pez, A.-M., Pelachaud, C.: Statistical gesture models for 3D motion capture from a library of gestures with variants. In: *Post-proceedings of the 8th International Gesture Workshop (GW 2009)*, Bielefeld, Germany. LNAI, vol. 5934, Springer, Heidelberg. To appear.
22. Allusse, Y., Horain, P., Agarwal, A., Saipriyadarshan, C.: GpuCV: An OpenSource GPU-Accelerated Framework for Image Processing and Computer Vision. In: *Proc. ACM Multimedia 2008 Open Source Software Competition*, Vancouver, BC, Canada, pp. 1089-1092, ACM (2008). [doi:10.1145/1459359.1459578].
23 Allusse, Y., Horain, P., Agarwal, A., Saipriyadarshan, C.: GpuCV: A GPU-accelerated framework for Image Processing and Computer Vision. In: *Advances in Visual Computing*, *Proceedings of the 4th International Symposium on Visual Computing (ISVC08)*, Las Vegas, Nevada, USA. LNCS, vol. 5359, pp. 430-439. Springer Berlin / Heidelberg (2008) [doi:10.1007/978-3-540-89646-3_42].
24. Willow Garage: *OpenCV*, http://www.willowgarage.com/pages/software/opencv.
25. Second Life, *What is Second Life?*, http://secondlife.com/whatis.