ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Graphics 29 (2005) 179–187
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

Hands and face tracking for VR applications

Javier Varona*, José M. Buades, Francisco J. Perales

Unidad de Gráficos y Visión por Ordenador, Dept. de matematiques i Informatica, Universitat de les Illes Balears (UIB),
crta. Valldemossa km. 7,5, 07122 Palma de Mallorca, Spain

Abstract

In this paper, we present a robust real-time 3D tracking system of human hands and face. This system can be used as a perceptual interface for virtual reality activities in a workbench environment. The main advantage of our system is that the human, placed in front of the virtual reality device, does not need any type of marker or special suit. The system includes a colour segmentation module to detect in real-time the skin-colour pixels present in the images. The results of this skin-colour segmentation will be skin-colour blobs, these are the inputs of a data association module. This module labels the blobs pixels using a set of hypothesis from previous frames. The 2D-tracking results are used for the 3D reconstruction of hands and face in order to obtain the 3D positions of these limbs. Finally, we present several results using the H-ANIM standard to show the system's output performance.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Visual tracking; Motion capture; Perceptual user interfaces; Virtual reality

1. Introduction

In order to allow a user to navigate in a 3D-space, most commercial systems encumber the user with head-mounted displays, electro-magnetic or ultrasound position sensors, gloves and/or body suits [1]. Although such systems can be extremely accurate, they limit the freedom of the user due to the tethers associated with the sensors and displays. We have, therefore, chosen to build a vision system to obtain the necessary detailed information about the user. We have specifically avoided solutions that require invasive methods like special clothing or special hardware.

Our interactive 3D-space is a desktop that consists of a workbench with two projection screens, see Fig. 1. The 3D-space is instrumented with a camera stereo pair. The stereo pair is used for visual tracking human movements. This configuration allows the user to view a virtual world while standing in front of the workbench. Gesture and manipulation occur in the workspace defined by the screens and user.

Many preceding works developed in this topic already exist, for example, finger detection with multi-scale colour approach [2], or in real-time for virtual reality applications in 2D [3], and for human grasping using infrared images [4]. Instead of the techniques used in these preceding works, we would like to define a general, robust and efficient system that can be used with non-expensive cameras (using day light spectrum) and that it can retrieve inputs coming from a human interaction with a virtual reality system without any type of marker or special suit.

The system must detect a new user entering into the system's environment and analyse him to set parameters such as his skin-colour. Once the user who is going to interact with the machine is detected, the system starts tracking interesting regions such as the extreme limbs

*Corresponding author. Fax: +34 971173003.
E-mail address: vdmijvg4@uib.es (J. Varona).

0097-8493/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2004.12.002

Fig. 1. Interactive 3D-space.

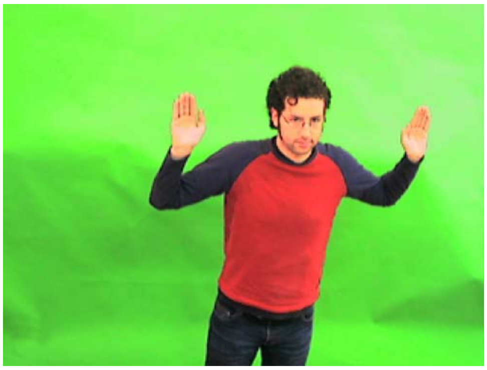

Fig. 2. Initial user posture for selecting the skin-colour samples.

(i.e. face and hands), using information obtained in the user detection task. The input data for the gesture interpretation process are the position and orientation of these regions. This process will recognize which gesture the user has carried out. Next, these gesture data are sent to the execution process, which ends the process by performing the action that has been specified and completing the feedback process. This is a very complex and challenging task, so we isolate sub problems to make it more feasible.

In particular, the whole system must be able to work with a complete kinematical model of the human body. That means that the final 3D reconstruction will be for all the visible joints and segments in the application domain. In the case of virtual reality applications, the upper body is the kinematical chain assumed to be important. From our system, we try to detect and track the end-effectors (hands and face) and their kinematical structure. Also, we plan to apply priorized inverse kinematics [5] for recovering the other degrees of freedom of the human upper body. This kind of IK can be controlled by biomechanical subtasks and a set of rules from image segmentation restrictions. In this paper, we will then make some assumptions to simplify the problem and make it near real time.

In the following sections, we explain briefly the proposed tracking method including the skin-colour pixel detection and training. We also present our 2D tracking and 3D reconstruction processes and finally we conclude with some visualization results in VRML format and H-Anim avatar compliant [8].

2. Hands and face tracking algorithm

In our case, the tracking problem lies in identifying both hands and face in each image of the input stereo pair. In order to do this, we should detect skin-colour pixels in the image to build a blob-based representation [9] of each extreme limb candidate (in advance, we refer the hands and the face as extreme limbs). Next, we use a data association algorithm that relates the detected skin-colour blobs with a set of hypothesis built from the extreme limbs states at the previous frame. For calculating these hypotheses, we apply a simple prediction scheme to the detected extreme limbs. Finally, we obtain the 3D positions of the extreme limbs triangulating their 2D image positions in both cameras.

2.1. Skin-colour segmentation module

The assumption that colour can be used as a cue to detect faces and hands has been proved in several publications [10,11]. Then, when a new pair of frames arrive, the first step is to detect the skin-colour pixels in the image. We have another method based in the Mumford–Shah functional [6,7], but it has high computational cost. An easier solution that performs well in real-time is based on a probabilistical modelling of the skin-colour pixels distribution.

Therefore, before applying the skin-colour detection, it is necessary to model the actor's skin-colour in a previous step. In our work, we only use one image of the actor for making this model. From an initial actor's posture, see Fig. 2, the system locates the regions of the image that contain skin-colour pixels. Next, we transform these pixels from the RGB-space to HSL-space to take the chroma values for each pixel, that is, the hue and saturation values form a skin-colour sample.

The hue and saturation values contain the chroma information, however, using them for skin-colour segmentation cause two main problems. On the one hand, human skin hue values are near the red colour, that is, they have a value near 2π radians. In this case, it is difficult to learn the distribution because the hue angular nature can produce samples at both limits. Therefore, to solve this problem we rotate the hue values π radians before selecting each sample. Then, we can use standard statistical models to learn the skin-colour distribution.

On the other hand, when saturation values are near 0, the hue is unstable, and this can cause false detections for example, for black and white colours (it is clear that in this case there is no chroma information). Therefore, when we are selecting the skin-colour samples, we discard saturation values near to 0.

Once these two preprocessing steps for the selected samples are completed, we have a sample set to learn the skin-colour distribution

$$X = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n\},$$

where n is the number of samples and each sample i is composed of the hue and saturation values, $\mathbf{x}_i = (h_i, s_i)$. As a result of a testing and comparing phase with several statistical models such as mixture of gaussians or discrete histograms, the best results have been obtained using a Gaussian model, that is

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i, \tag{1}$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n} \langle (\mathbf{x}_i - \overline{\mathbf{x}}), (\mathbf{x}_i - \overline{\mathbf{x}})' \rangle. \tag{2}$$

Once we find the values of the parameters $\overline{\mathbf{x}}$ and Σ using the sample set, we can calculate the probability that a new pixel is skin

$$P(\mathbf{x}\,\text{is skin}) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} \times \exp\left(-\tfrac{1}{2}\langle (\mathbf{x} - \overline{\mathbf{x}})|\Sigma^{-1}|(\mathbf{x} - \overline{\mathbf{x}})' \rangle\right). \tag{3}$$

In Fig. 3 several results of this process are shown, it can be seen in these images how this model performs well in several lighting conditions.

We obtain the blob representation of each extreme applying a connected components algorithm [12] to the probability image, which groups pixels into the same blob. In Fig. 4 we show the final results of skin-colour detection.

The final step of the skin-colour detection is the computation of the attributes of each blob (centroid, size and orientation) that will be used by the data association process to represent the state of the actor's extreme limbs.

2.2. Data association module

The objective of the data association is to propagate in time the labels that represent the face, the left hand

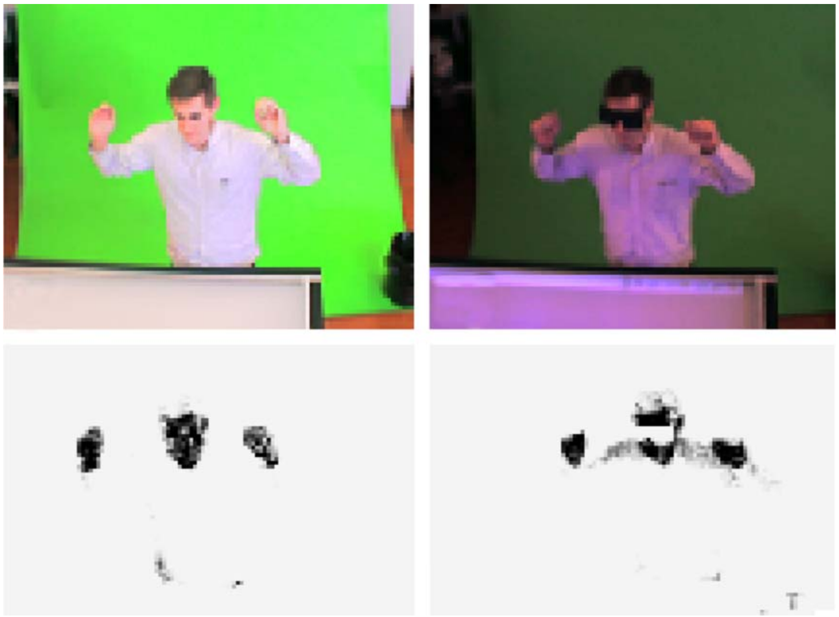

Fig. 3. Skin-colour probability images for different lighting conditions.

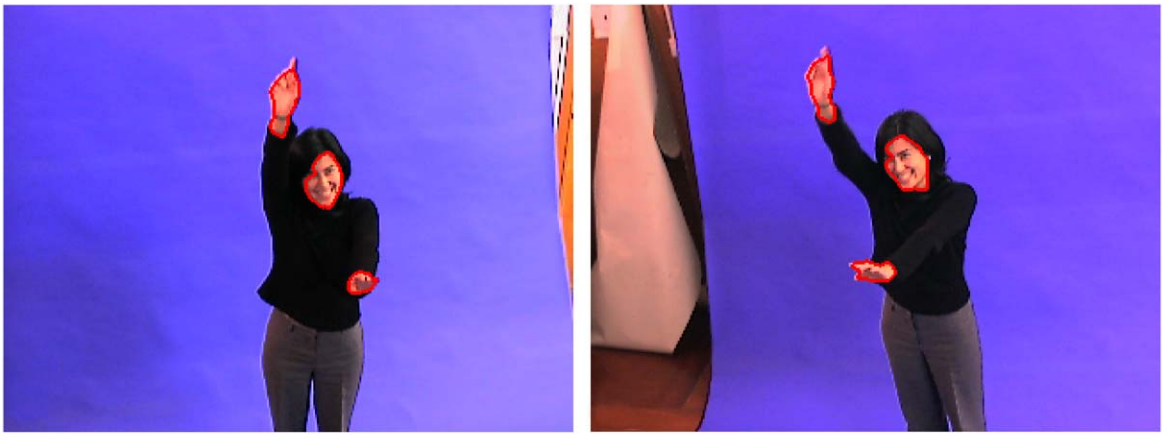

Fig. 4. Contours of skin-colour blobs after the connected components process.

and the right hand, and to detect when a new extreme limb enters in the 3D-scene or disappears from the field of view.

As we have seen before, we represent the state of the l extreme limb, $\mathbf{s}_l$, by means of a vector that contains the position, the size and the orientation of the extreme limb

$$\mathbf{s}_l = (\mathbf{p}_l, \mathbf{w}_l, \theta_l),$$

where l is the extreme limb label, $\mathbf{p} = (p_x, p_y)$ is its position in the 2D image, $\mathbf{w} = (w, h)$ is the size of the limb in pixels, and θ is its angle in the 2D image plane.

The data association algorithm operates as follows: for a frame at time t the goal is to associate the skin-colour blobs, found by the skin-colour detection procedure, with the extreme limb state hypothesis built from the state at time $t-1$. To do this, the extreme limb positions in time $t-1$ are propagated to the next frame t using the next linear scheme of prediction:

$$\hat{\mathbf{p}}(t) = \mathbf{p}(t) + \Delta\mathbf{p}(t-1), \tag{4}$$

$$\Delta\mathbf{p}(t) = \mathbf{p}(t) - \mathbf{p}(t-1). \tag{5}$$

The above equations express that an extreme limb will maintain the same velocity on the image plane.

Therefore, on the one hand, we have a set of hypothesis at time t

$$H = \{\mathbf{h}_l\}, \quad l \leqslant 3,$$

where

$$\mathbf{h}_l = (\hat{\mathbf{p}}_l, \mathbf{w}_l, \theta_l).$$

On the other hand, we assume that at time t, M blobs have been detected

$$B = \{b_1, \ldots, b_j, \ldots, b_M\},$$

where each blob with label b_j, correspond to a set of connected skin-colour pixels. Note that a blob may correspond to one or more extreme limbs. For example, two crossing hands are two different extreme limbs that appear as one blob when one occludes the other.

The data association process has to set the relation between hypothesis $\mathbf{h}_l$ and observations b_j in time. In order to cope with this problem, we define an approximation to the distance from the $\mathbf{x} = (x, y)$ image pixel to the hypothesis $\mathbf{h}$. First, we normalize the image pixel coordinates

$$\mathbf{t} = \mathbf{x} - \mathbf{p}, \tag{6}$$

$$\mathbf{n} = \mathbf{R} \cdot \mathbf{t}', \tag{7}$$

where

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}.$$

Then, by calculating the angle between the normalized image pixel and the hypothesis centre

$$\alpha = \operatorname{atan}(n_y/n_x),$$

we can find the crossing point, $\mathbf{c} = (c_x, c_y)$, between the hypothesis ellipse and the normalized image pixel as follows:

$$c_x = w\cos\alpha, \tag{8}$$

$$c_y = h\sin\alpha. \tag{9}$$

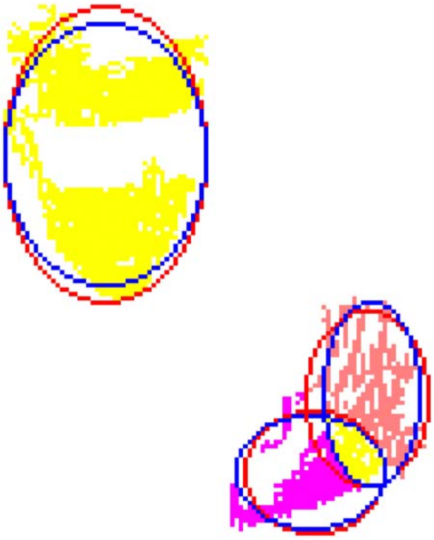

Fig. 5. Occlusion case solved using multiple labelling (object hypothesis are depicted using ellipses).

Finally, the distance between an image pixel and a hypothesis is

$$d(\mathbf{x},\mathbf{h}) = \|\mathbf{n}\| - \|\mathbf{c}\|. \tag{10}$$

From the distance definition of Eq. (10) it turns out that its value is equal to or less than 0 if $\mathbf{x}$ is inside the hypothesis $\mathbf{h}$, and greater than 0 if it is outside.

Therefore, considering a hypothesis $\mathbf{h}$ and a point $\mathbf{x}$ belonging to a blob b, if the distance is equal to or less than 0, we conclude that the blob b supports the existence of the hypothesis $\mathbf{h}$ and that this hypothesis predicts blob b. Now, if a new extreme limb appears in the 3D-scene implies that none of the existing hypothesis predict the existence of the corresponding blob b,

$$\forall \mathbf{x} \in b, \quad \min_{\mathbf{h}\in H}\{d(\mathbf{x},\mathbf{h})\} > 0. \tag{11}$$

Eq. (11) describes a blob with empty intersection with all hypotheses. Algorithmically, at each time instant t, all detected blobs are tested against the criterion of Eq. (11). If a blob does not belong to any hypothesis, a new extreme limb hypothesis is created and its parameters correspond to the blob parameters.

After the detection of new extreme limbs, all the remaining blobs must support the existence of limb hypothesis. Therefore, if a pixel $\mathbf{x}$ of a blob is inside a limb hypothesis then this pixel is labelled with the hypothesis number. Formally,

$$\forall \mathbf{x} \in B, \ \ \mathbf{x} = l \quad \text{iff} \quad d(\mathbf{x},\mathbf{h}_l) \leqslant 0. \tag{12}$$

Eq. (12) allows the treatment of limbs occlusion because the pixels belonging to a blob can be labelled with more than one hypothesis, see Fig. 5. The remaining pixels of a blob b that already has labelled pixels by Eq. (12) are assigned to the hypothesis number that is closer to it. Formally,

$$\forall \mathbf{x} \in b, \ \ \mathbf{x} = l \quad \text{iff} \quad \text{argmin}_l\{d(\mathbf{x},\mathbf{h}_l)\}. \tag{13}$$

Another interesting case can happen when the occlusion finishes and then an hypothesis is supported by more than one blob (split case). In this case, the hypothesis is assigned to the blob with which it shares the largest number of pixels.

Finally, a limb hypothesis should be removed when the limb moves out of the scene. Following our algorithm, a hypothesis $\mathbf{h}_i$ is removed when

$$\forall \mathbf{x} \in B, \quad \{d(\mathbf{x},\mathbf{h}_l)\} > 0. \tag{14}$$

In Fig. 6, we show a tracking example for the camera stereo pair. In this figure, we label the different extreme limbs using the red, green and blue colours. Also, the hypothesis ellipse is depicted using the magenta colour.

2.3. 3D-point reconstruction

Once we have the extreme limbs 2D positions in each image of the stereo pair, we can estimate their 3D position using standard geometry algorithms [13]. The 3D position is computed for each extreme limb, projecting the centroid of the blob on each image to infinity and computing its 3D coordinates as the nearest point to these two lines. However, in order to carry out the 3D point reconstruction process, it is necessary to calibrate the camera stereo pair.

For calibrating the cameras, we apply the Zhengyou Zhang algorithm [14] that uses a planar pattern as a calibration object (cheap and easy to use), see Fig. 7. This algorithm has the great advantage that it is implemented with minor differences by Jean–Yves Bouguet in MATLAB [15] and also in C + + code in the OpenCV Intel's library [16].

Finally, the complete tracking procedure for the 3D reconstruction process of hands and face is described in Fig. 8.

3. Visualization using H-Anim

The hands and face tracking system obtains 3D coordinates from the camera stereo pair. This information is needed to display the data in real time on the

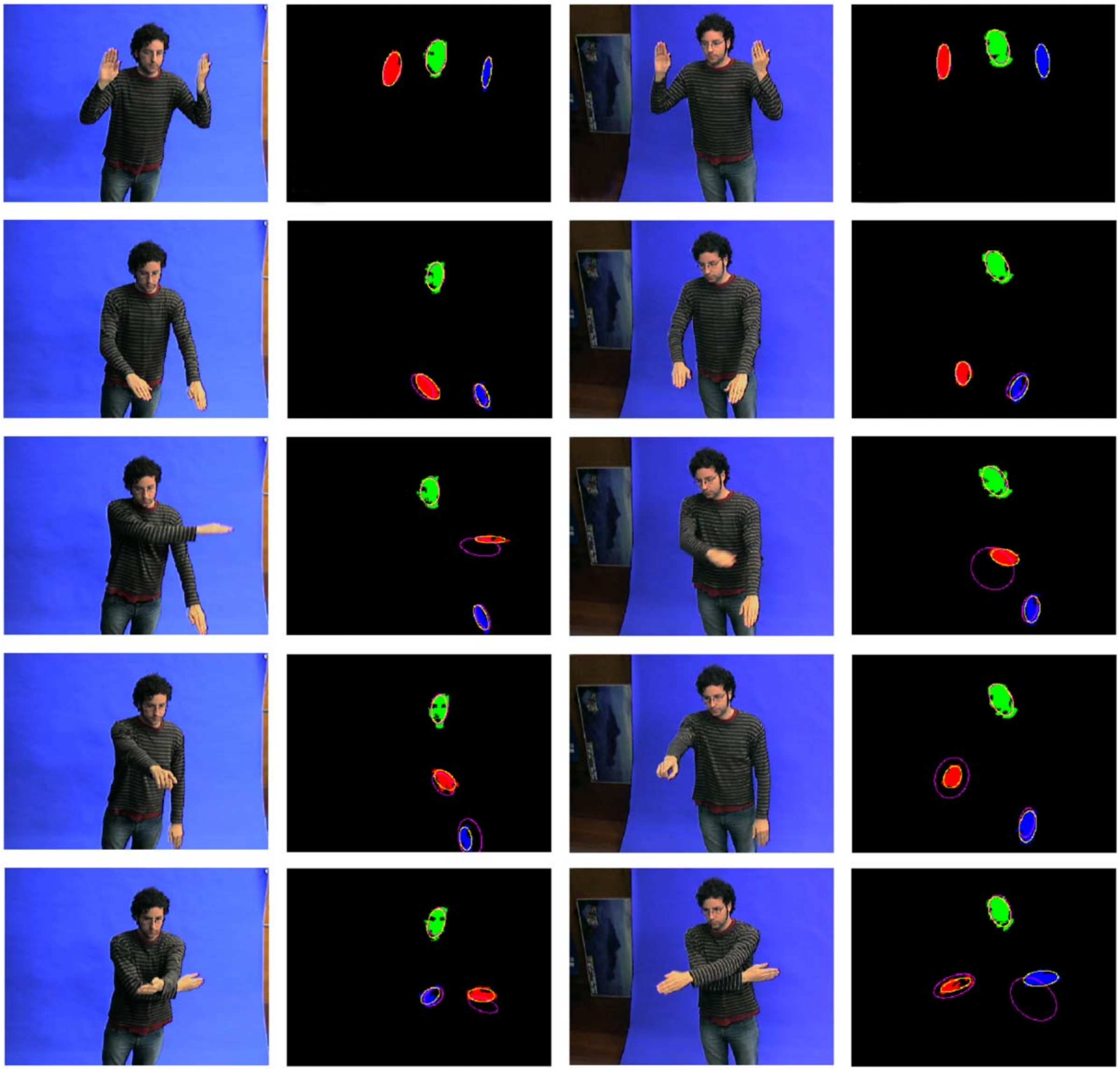

Fig. 6. 2D-tracking results. Left: Camera 1; Right: Camera 2.

workbench. That means that a human virtual avatar can be reproduced in a remote workbench. Then, for the human model representation we use the H-Anim standard, this way we can collaborate with standard VRML models (see Fig. 9).

However, the final objective of this process is to recover accurately the 3D positions and orientations of hands and face in the interactive applications. Therefore, hands and head are modelled as an H-Anim humanoid (see Fig. 10).

Actually, at the moment only 3D positions are computed, so we should consider the 3D orientation for completing the 3D reconstruction of the avatar in the future.

4. Conclusion and future work

In this paper, we have proposed a new system for 3D tracking of human extreme limbs for HCI in real time. The algorithm analyses the user for setting parameters that will be useful for the tracking process, for example his skin-colour. Next, a segmentation

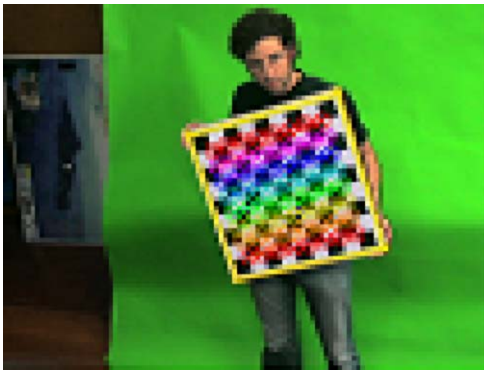

Fig. 7. Calibration procedure with a chessboard.

process based on the pixel colour and a hypothesis-based data association algorithm to track the user's hands and face are defined. Besides, the whole process is carried out in real time.

The results of this process are the face and hands 3D tracking and their 3D reconstruction to allow an human–computer interaction system for virtual reality environments using the standard H-Anim. Also, the system can be used in distributed virtual environments because we do not need to send the real images of the human body. We only need to send the 3D positions and orientations of the model recovered by our tracking system.

It remains as future work, the tracking of interesting body parts (all the upper torso) and the understanding

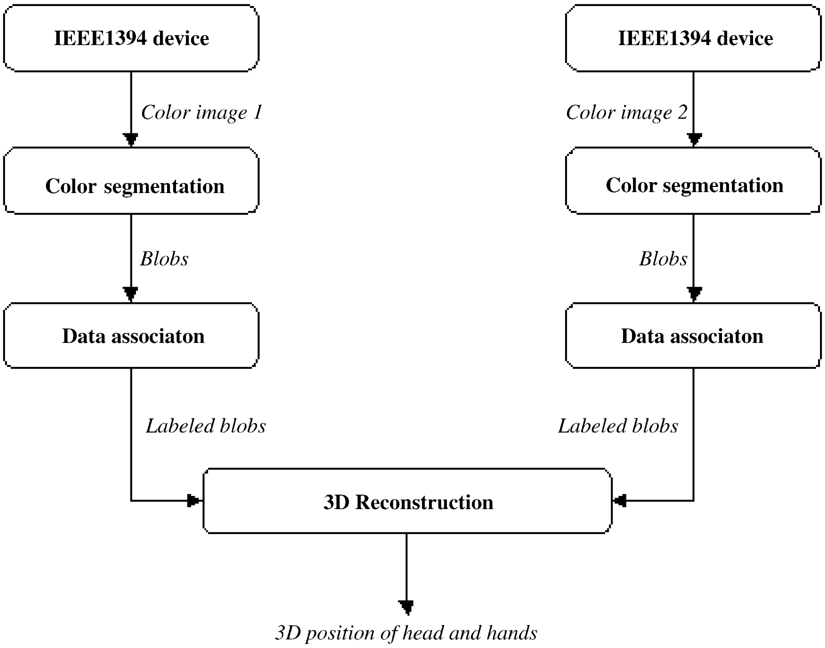

Fig. 8. Complete procedure: color segmentation, data association and 3D reconstruction.

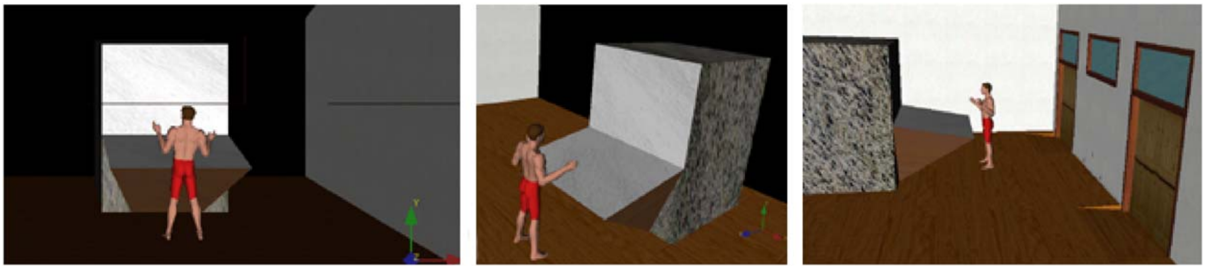

Fig. 9. Different views of the reconstruction using H-Anim.

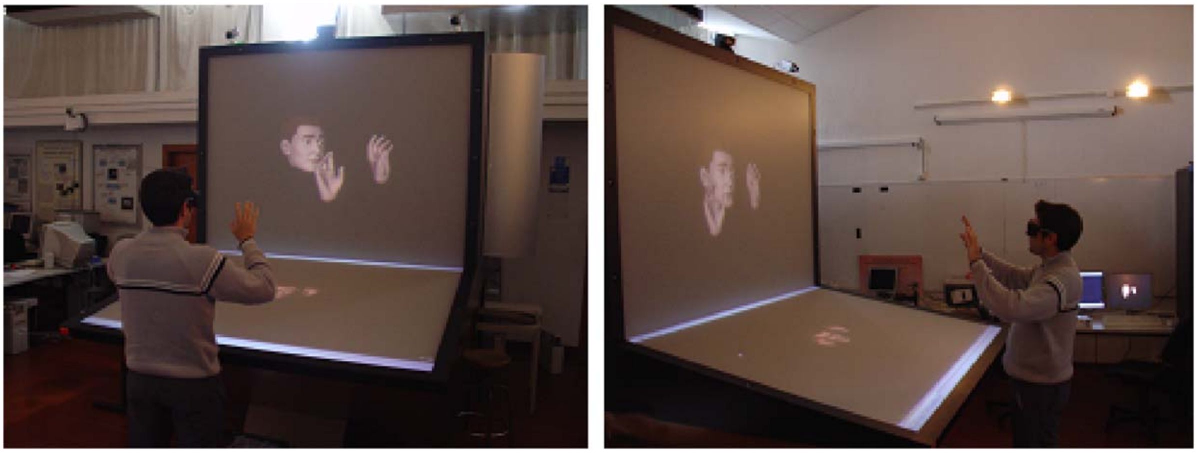

Fig. 10. An example of HCI in virtual reality applications.

of movements in order to carry out the gesture recognition that the user is performing. On the other hand, we must improve the results, in particular we need to compare our system with commercial trackers to evaluate the true accuracy.

Acknowledgements

The projects TIC2003-0931 and TIC2002-10743-E of MCYT Spanish Government and the European Project HUMODAN 2001-32202 from UE V Program-IST have subsidized this work. Also we would like to express our gratitude to all members of Computer Graphics and Vision group at UIB for their collaboration. Javier Varona acknowledges the support of a Ramony Cajal Fellowship from the Spanish ministerio de Education y Ciencia.

References

[1] Perales FJ. Human motion analysis and synthesis using computer vision and graphics techniques. State of art and applications. Proceedings of the world multiconference on systemics, cybernetics and informatics (SCI2001), 2001.

[2] Bretzner L, Laptev I, Lindeberg T. Hand gesture recognition using multiscale colour features, hierarchical models and particle filtering. Proceedings of the fifth IEEE international conference on automatic face and gesture recognition (FGR.02), 2002.

[3] Okay K, Satoy Y, Koikez H. Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems. Proceedings of the fifth IEEE international conference on automatic face and gesture recognition, 2002.

[4] Ogawara K, Hashimoto K, Takamatsu J, Ikeuchi K. Grasp recognition using a 3D articulated model and infrared images. Institute of Industrial Science, University of Tokyo, Tokyo, Japan, Fuji Xerox Information Systems Co., Ltd., Tokyo 1500031, Japan.

[5] Peinado M, Herbelin B, Wanderley M, Le Callennec B, Boulic R, Thalmann D, Méziat D. Towards configurable motion capture with prioritized inverse kinematics. Proceedings of the third international workshop on virtual rehabilitation (IVWR04), Lausanne, 2004.

[6] Buades JM, González M, Perales FJ. A new method for detection and initial pose estimation based on Mumford–Shah segmentation functional. Proceedings of IbPRIA 2003, Port d'Andratx, Spain, Lecture Notes in Computer Science, vol. 2652. Berlin: Springer; 2003.

[7] Buades JM, González M, Perales FJ. Face and hands segmentation in colour images and initial matching. International workshop on computer vision and image analysis, Palmas de Gran Canaria, Spain, 2003.

[8] HANIM 1.1 Compliant VRML97. http://ece.uwaterloo.ca/h-anim/index.html.

[9] Wren C, Azarbayejani A, Darrell T, Pentland A. Pfinder: Real-time tracking of the human body. IEEE Transactions Pattern Analysis and Machine Intelligence 1997;19(7): 780–5.

[10] Bradski GR. Computer video face tracking for use in a perceptual user interface. Intel Technology Journal, Q2'98, 1998.

[11] Comaniciu D, Ramesh V. Robust detection and tracking of human faces with an active camera. Proceedings of the third IEEE international workshop on visual surveillance, 2000. p. 11–8.

[12] Raimi A. Fast connected components on images. http://xenia.media.mit.edu/~rahimi/connected.

[13] Hartley R, Zisserman A. Multiple view geometry in computer vision. Cambridge: Cambridge University Press; 2000.

[14] Zhang Z. Flexible camera calibration by viewing a plane from unknown orientations. Proceedings of ICCV'99 1999;(1):666–73.

[15] Bouget JY. Camera Calibration Toolbox. http://www.vision.caltech.edu/bouguetj.

[16] Intel OpenCV library. http://www.intel.com/research/mrl/research/opencv/index.htm.

Javier Varona has been granted with a "Ramon y Cajal" research fellowship of the Spanish Government in the Universitat de les Illes Balears (UIB). He received his Ph.D. degree in Computer Engineering from Universitat Autònoma de Barcelona (UAB) and the Computer Vision Center (CVC), in 2001. His Ph.D. thesis research was on robust visual tracking. Next, he arrived at the UIB to take part in the European research project HUMODAN (IST-2001-32202). His research interests include 3D-tracking, human motion capture and face tracking.

José M. Buades is an assistant professor at the UIB (Balearic Island University), Spain, where he carries out his research tasks. His research interests include virtual environments, augmented reality, human–computer interaction and rendering in OpenGL. He is doing his Ph.D. in human–computer interaction area, focused on initialization and user detection.

Francisco J. Perales is an expertise in the area of computer graphics and computer vision, obtained his BS and MS in computer science at Universidad Autonoma de Barcelona (UAB) in 1986. He also obtained a Master of Digital Image Processing at UAB in 1993. Finally, he received the Ph.D. degree in Computer Science at the University of Balearic Islands in 1993. He is also working for a Ph.D. in medical sciences applied to 3D multimodal image processing. He is currently a professor in the Department of Mathematics and Computer Science of UIB. Dr. Francisco J. Perales got his thesis titled "Human Motion Analysis using digital images processing". Since then, his main area of work is the analysis, synthesis and understanding human movements and deformable objects modelling. He is the leader of several projects on Human Motion Analysis & Synthesis founded by the Spanish Government (TIC98-0302, TIC2001-0931). Also, he is the Scientist in Charge of HUMODAN IST-2001-32202 (An automatic human model animation environment for Augmented Reality interaction) European project at UIB. Also, in our Math and Science Department we have a working group in computer graphics and vision and Dr. Francisco J. Perales is the actual co-ordinator.