Toward natural interaction through visual recognition of body gestures in real-time

Javier Varona [a,*], Antoni Jaume-i-Capó [a], Jordi Gonzàlez [b], Francisco J. Perales [a]

[a] *University of the Balearic Islands, Department of Mathematics and Computer Science, Palma de Mallorca, Spain*

[b] *Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Barcelona, Spain*

Abstract

In most of the existing human-computer interfaces, enactive knowledge as new natural interaction paradigm has not been fully exploited yet. Recent technological advances have created the possibility to enhance naturally and significantly the interface perception by means of visual inputs, the so-called Vision-Based Interfaces (VBI). In the present paper, we explore the recovery of the user's body posture by means of combining robust computer vision techniques and a well known inverse kinematics algorithm in real-time. Specifically, we focus on recognizing the user's motions with a particular mean, that is, a body gesture. Defining an appropriate representation of the user's body posture based on a temporal parameterization, we apply non-parametric techniques to learn and recognize the user's body gestures. This scheme of recognition has been applied to control a computer videogame in real-time to show the viability of the presented approach.

Key words: Enactive Interfaces, Human-computer interaction, Vision-based interfaces

1 Introduction

Enactive knowledge represents the kind of knowledge *learned by doing*, based on the experience of perceptual responses to action, acquired by demonstration and sharpened by practice. Although until now human-computer interaction technologies have not fully exploited the potential of enactive knowledge, recent technological advances have created the possibility to significantly enhance the interface perception by means of visual inputs, the so-called Vision-Based Interfaces (VBI) proposed by Turk and Kolsch (2004).

Vision-based interfaces use computer vision in order to sense and perceive the user and their actions within an HCI context. Computer Vision technology applied to the human-computer interface has notable success to date (Moeslund et al., 2006a). From a human-computer interaction point of view, we are especially interested in obtaining user motions in order to recognize those that can be interpreted as system's events. In this sense, the approaches used for recognition and analysis of human motion in general can be classified into three major categories: motion-based, appearance-based, and model-based approaches. Motion-based approaches attempt to directly recognize the gestures from the motion without any structural information about the physical body (Bobick and Davis, 2001; Efros et al., 2003). Appearance-based approaches use two di-

* Corresponding author. Address: Edificio Anselm Turmeda Campus UIB, Cra. Valldemossa, km 7.5, 07122 Palma de Mallorca, Spain. Tel.: +34 971 172005; fax: +34 971 173003.

Email address: xavi.varona@uib.es (Javier Varona).

mensional information such as gray scale images, edges or body silhouettes (Elgammal et al., 2003). In contrast, model-based approaches focus on recovering the three dimensional configuration of articulated body parts (Ren et al., 2004; Kojima et al., 2000).

It is clear that recovering the user posture should be more useful than the other approaches as it gives a complete description of the user motions in 3D. However, model-based approaches are often difficult to apply to real-world applications. This fact is mainly due to the difficulty of capturing and tracking the requisite model parts, the user's body joints that take part in the considered gestures. Besides, in order to use this approach for interaction, the algorithms must work in real-time and the majority of model-based approaches perform in an off–line fashion. A partial solution is to simplify the capture by reducing the number of body parts and using its temporal trajectories in order to recognize the gestures of interest (Wu and Huang, 1999). For example, Rao et al. (2002) analyze the problem of learning and recognizing actions performed by a human hand. They target affine invariance and apply their method to real image sequences using skin color to find the hands. They characterize a gesture through dynamic moments, which they define as maxima in the spatio-temporal curvature of the hand trajectory that is preserved from 3D to 2D. Their system does not require a model; in fact, it builds its own model database by memorizing the input gestures. Other approaches of hand-based gesture recognition methods use hand poses as gestures for navigating in virtual worlds (O'Hagan et al., 2002). Nevertheless, exploiting the sole 3D location of one or two hands is indeed not sufficient to recognize complex gestures in order to control interactive applications. Instead of proposing another partial solution, this paper presents a model-based approach founded on

the user's posture recovery in real-time. Our approach presents a vision-based system to obtain user's motions through a combination of the analysis of the images provided by two cameras (observation) and a real-time implementation of a known inverse kinematics algorithm (control). This system combines video sequence analysis and visual 3D tracking to deliver the user's motions in real-time. This allows the end user to make large upper body movements naturally in a 3D scene.

In addition, the system is able to process, not only the 3D position of the user's joints, but also to report a set of body gestures and hence offering a richer user interface. We define as a capable gesture recognition system when a gesture of interest is recognized to generate the desired computer event in real-time. To achieve this objective, we address the main problems in the gesture recognition challenge: temporal, spatial and style variations between gestures. Temporal variations are due to the difference in the speed of gestures between different users. Spatial variations are due to physical constraints of the human body such as different body sizes. Style variations are due to the personal way in which users makes their movements. To cope with spatial variations we normalize the computed joints positions. Temporal variation is managed using a temporal gesture representation. Finally, the most difficult challenge, style variations, are solved using a non-parametric scheme for learning and recognition.

In order to show the viability of this scheme of recognition, an enactive interface to control a computer videogame in real-time has been developed. With this application, the user's body acts as a new device to interact with the computer showing its adaptive flexibility to the particular way of creating the gestures of each user. This simple example opens a rich potential of intu-

itive manipulation of entities through the presented approach in new complex scenarios of natural interaction.

This paper is organized as follows. The real-time full-body motion capture system used to obtain the user's motions is presented in section 2. Next, in section 3, our motion recognition approach is described, that is, how the recognition challenges explained above are solved. The application of our system in a real-time interactive application and the obtained results are described in section 4. The obtained results are discussed in the last section to demonstrate the viability of this approach.

2 Obtaining user motions

This section describes the proposed methods used to obtain user's motions in real-time. Our system is based on the combination of visual cues and inverse kinematics (IK). Therefore, the images from two synchronized colour cameras represent the input of the system. Usually, these images can be noisy or incomplete (some joints or limbs aren't visible). Therefore, we can only estimate the user's posture. IK approaches can solve the body posture from known positions of the end-effectors (hands and face for the upper body case). We propose a scheme where these end-effectors are automatically located in real-time and fed into a robust algorithm of Inverse Kinematics. This algorithm allows the definition of a set of constraints to guide the estimated user's posture toward plausible balanced human body configurations in few convergence steps to ensure a real-time response.

2.1 Visual cues

For each instant in time, we must locate the user's end-effectors in each image. We use skin-color segmentation, 2D-tracking and 3D-tracking algorithms to estimate the 3D positions of both hands and face in the scene. First, we use a skin-color detection module to find the skin-color pixels present in the images. The results of this skin-color detection will be skin-color blobs, which are the input of a 2D-tracking module. This module labels the blob's pixels using a hypothesis set from previous frames (Varona et al., 2005).

Once we have the end-effectors' 2D positions in each image of the stereo pair, we can now estimate their 3D position using the mid-point triangulation method. With this method, the 3D position is computed projecting each end-effector 2D position to infinity and subsequenlty taking the nearest point to these two lines (Trucco and Verri, 1998). However, in order to execute this 3D point reconstruction process, an extra computational step is required, which will robustly relates to the stereo pair measurements of the end-effectors. In the case of severe occlusion, the end-effectors labels do not agree in both images. The result is that the 3D point reconstruction for these limbs is not correct. However, since the positions of the end-effectors are in the 3D world, we can use a physical model to track them. A limb in time t is characterized by its position, which is represented by a state vector $\mathbf{x}_t$. The imaging system observes the projected limb 3D position in the vector $\mathbf{z}_t$ (i.e. the triangulated position from the two different views). The limb's dynamics is assumed to be described by the difference equation:

$$\mathbf{x}_t = \mathbf{f}_{t,t-1}(\mathbf{x}_{t-1}) + \mathbf{w}_t, \tag{1}$$

where $\mathbf{f}_{t,t-1}(\cdot)$ is a vector function describing the transition of the state vector from $t-1$ to t, and $\mathbf{w}$ represents the error model. The state transition function for a limb is a kinematics polynomial model assuming constant velocity. The measurement equation describes the relation between the observed positions and the state variables of the dynamic system:

$$\mathbf{z}_t = \mathbf{m}_{t,t-1}(\mathbf{x}_t) + \mathbf{n}_t, \tag{2}$$

where $\mathbf{m}_{t,t-1}(\cdot)$ is the measurement function and $\mathbf{n}$ is the measurement noise. The Kalman filtering equations allow computing the optimal estimates of the state vector recursively from the measurements and the initial estimation. In order to do this, we first triangulate all the possible combinations of 2D measurements from the two images to obtain the 3D position candidates of each end-effector. Subsequently for each end-effector we select the candidate nearest to the position predicted by the estimation filter. Figure 1 shows the results of this process by backprojecting the corrected associate end-effectors 3D position in the 2D images of the stereo pair after a severe occlusion.

2.2 User's body model and adjustment

Due to our interest in the posture recovery for interaction purposes, we use an articulated body model with 15 degrees of freedom that is enough to analyze the user's motions. Specifically, our user's body model consists of a Virtual foot (2 dofs), that roots the body to the floor with frontal and lateral axes of rotation, a Back (2 dofs), that corresponds to the beginning of the spine with frontal and lateral axes of rotation, the Thorax (3 dofs), which has all the rotation axes, the Shoulders (2×3 dofs) and the Elbows (2×1 dof), see

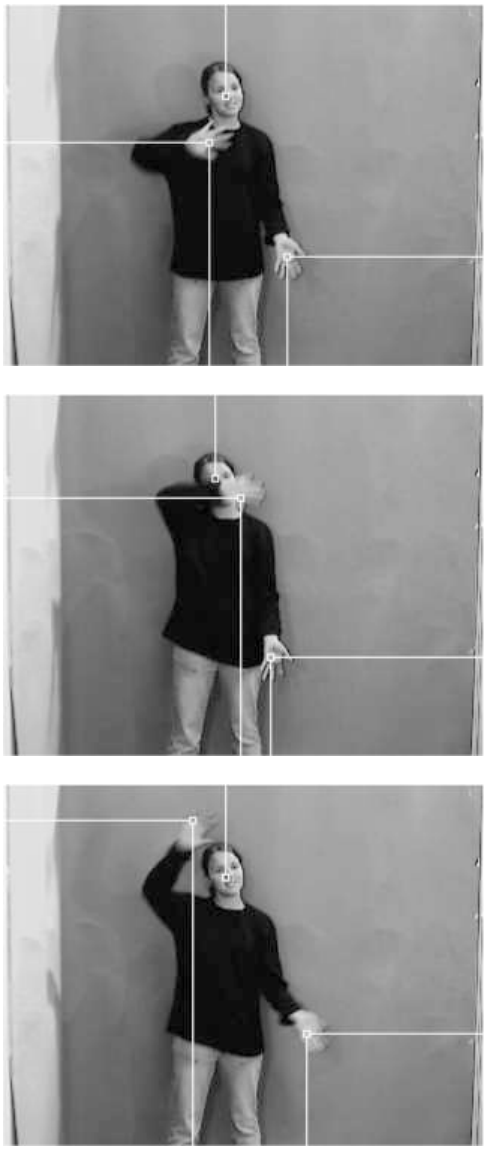

Fig. 1. Corrected end-effectors tracked 3D positions backprojected in both images (up from left camera and down from right camera). The white line starting in the right boundary image corresponds to right hand and vice versa.

Figure 2. We use an initial manual joint location of the shoulders, the elbows and the hands for computing the lengths of the segments that remain constant for the rest of the session. We can derive the location of the other joints as a relative proportion of the lower body segment and the back segment, which are considered constant.

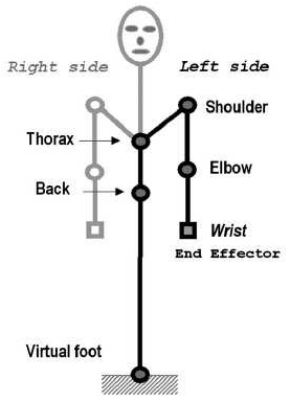

Fig. 2. Human body model.

As explained before, we apply the computer vision algorithms to obtain the 3D measurements of the user end-effectors. However, locating all the user body joints to recover the posture is not possible with only computer vision algorithms. This is mainly due to the fact that most of the joints are occluded by clothes. Therefore, if we can clearly locate visible body parts (the hands), Inverse Kinematics approaches can solve the body posture from its 3D position. We propose a scheme where the hands are automatically located in real-time and fed into an Inverse Kinematics module which in turn can provide a 3D feedback to the vision system.

The multiple Priority IK (also called Prioritized IK, or PIK) is exploited to reconstruct an anatomically correct posture of the user (i.e. its joint state, θ) using the 3D location of selected end-effectors (noted $\mathbf{x}$) measured with the vision system and used to constrain the posture. The PIK algorithm is based on the linearization of the set of equations expressing Cartesian constraints $\mathbf{x}$ as functions of the joints' degrees of freedom θ. We denote $\mathbf{J}$ the Jacobian matrix and use its pseudo-inverse, noted $\mathbf{J}^+$, to build the projection operators on the kernel of $\mathbf{J}$, noted $N(\mathbf{J})$. Our approach relies on an efficient computation of projection operators that allow splitting the constraints set into multiple

constraint subsets associated with a strict individual priority level (Baerlocher and Boulic, 2004). The solution guarantees that a constraint associated with a high priority will be achieved as much as possible while a low priority constraint will only be optimized only on the reduced solution space that does not disturb all higher priority constraints.

Hence, it is very important to identify which constraint has the higher impact on the quality of the convergence and the visual appearance of the reconstructed posture. As we address the posture recovery of a standing person, the believability of the recovered posture is mostly governed by the correctness of its balance. For these reasons, we propose to exploit a skeleton able to model a simplified mass distribution of the whole body and to offer a control of the whole body centre of mass. The prior observations on believability and reachability lead us to assign the highest priority to the centre of mass position constraint: this constraint ensures that the centre of mass projects over the root node (the virtual foot in Figure 2) to guarantee balance. Subsequently, the next most important constraint is the hand position recovered through the vision system. Immediately under the hand constraint we activate two low level constraints respectively on the shoulders (attracted to the initial location in space that were obtained at the calibration stage) and on the elbow (attracted towards their lowest possible position to produce a more natural posture).

2.3 Performance evaluation

The system has been implemented in Visual C++ using the OpenCV libraries (Bradski and Pisarevsky, 2000) and it has been tested in a real-time

interaction context on an AMD Athlon 2800 + 2.083 GHz under Windows XP. The images have been captured using two DFW-500 Sony cameras. The cameras provide 320×240 images at a capture rate of 30 frames per second. In our laboratory tests we have found that the system operates at 48Hz (24 fps for each camera) if we don't iterate the PIK. If we use 5 iterations the system's performance decreases to 22 fps and for a maximum of 20 iterations the system operates at 19 fps. These results ensure a real-time response of the system.

First, the computer vision algorithms are validated to measure the accuracy of the results of our algorithm, the end-effectors' 3D position. The 3D position is found by an ultrasound positioning device, the IS-900 MiniTrax Wireless Wand from the InterSense Company (InterSense Inc. Website, Last accessed 2008). In this experiment, the user holds the device with one hand, see Fig. 3. Then, we obtain the positions tracked by our system and the reported positions of the IS-900 device at the same time instants. With the two point sets in the same reference system, we apply as error measure the root mean-squared error (RMSE):

$$E = \frac{1}{n} \sum_{i} \|\vec{y_i} - \vec{z_i}\|. \tag{3}$$

where $\vec{z_i}$ is the 3D position tracked by our computer vision algorithms, and $\vec{y_i}$ is the 3D position detected by the ultrasound device. In order to make a thorough testing we perform a set of different experiments:

- Comparison of static key positions.
- Comparison of predefined movements ("moving arm").
- Comparison of short sequences of random movements.

Fig. 3. Configuration to evaluate computer vision algorithms.

Table 1

Evaluation results of the end-effectors 3D tracking.

Experiment	RMSE (in cm)	Num of frames
Static (Jitter)	0.48	376
Moving arm	1.24	116
Random movements (short)	4.03	849
Random movements (long)	5.43	2465

- Comparison of long sequences of random movements.

Table 1 shows the mean errors (in centimetres) obtained in different tests with different users for the four experiments. First, the experiment with a static position is useful to measure the jitter error from the two devices. As it is shown in Table 1, the jitter can be quantified in 5 millimetres (in fact, this value is the minimum accuracy reported by the InterSense ultrasound sensor). In the experiments, it can be viewed that the mean error grows and stabilizes in a maximum of 5.5 cm. This error is mainly due to the hand shape, that is, the hand is imaged from the cameras at different sizes and then the centre of gravity varies with the shape. This is the main deviation of the ultrasound

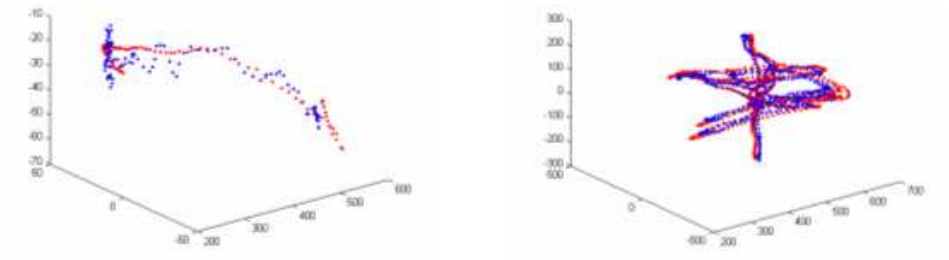

Fig. 4. Left: 3D trajectories of a predefined movement. Right: 3D trajectories of random movements. Ultrasound sensor in red, our system in blue.

device. In Fig 4, the tracks in 3D-space for the two positioning systems in two different experiments are shown. It can be viewed in these figures that the tracks are equal to some deviation due to the different hand shape imaged.

In order to evaluate the complete system including PIK, we test the application's results versus ground-truth using annotated sequences. We compare the elbows' positions between annotated points and our detected points. For comparison, positions of the elbows were chosen as they are the joints of the human upper body that move in these two scenes (without considering the end-effectors) and because their values are estimated by means of the combination between the vision-guided end-effectors tracking and the joint estimation from PIK. The first sequence has 450 frames corresponding to 15 seconds of real-time. In this sequence, human motions are smooth and there are no difficult occlusions between end-effectors that can distract the motion capture process. In this test, the mean error of the estimation of both elbows versus the ground truth data is similar and can be quantified around **5cm**. The second sequence is composed of 600 frames, corresponding to 20 seconds of real-time. In this sequence the user moves his arms freely without any constraint. The motions are fast and important end-effectors occlusions exist, for example when the user crosses his arms, see Fig. 5. In this case, the error

Fig. 5. Second test sequence. In this sequence the user moves his arms freely without any constraint, the motions are fast, and important end-effectors occlusions are noticed.

produced by both elbows is also similar and can be quantified around **12cm**. The error can be high if the performer raises his elbow up high because the PIK attracts the elbow downward as we assume this is more natural and we have no other information to control the elbow.

Finally, we also test our application performing several predefined arm motions and comparing the results with the desired final positions between motions. In order for the hierarchy to function correctly, the initial posture of the user's arm must be fully extended along the body so it can be determined the maximum extension of the arm. Firstly the user must flex one elbow until maximum flexion (this is not easy for inverse kinematics because the initial posture is singular); secondly, the centre of mass influence can be tested by using only one shoulder joint to move the arm laterally: when the arm is horizontal try to reach the furthest lateral point. This will force the user to counter balance the upper body posture with the lower body. How the elbow test and the centre of mass task work properly in these cases can be seen in Figure 6.

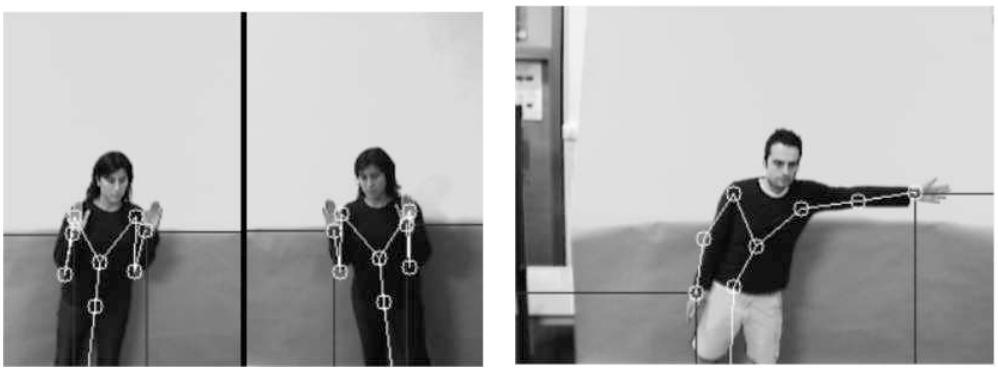

Fig. 6. Estimated postures for several predefined arm motions.

3 Recognizing motions

In general, we could classify the variations in which users perform their motions in three types: spatial, temporal and style variations. In this section, we explain how to cope with these types of variations. In order to make data invariant to different body sizes, the first step is to change the reference system because the calibration process of the Vision-PIK algorithm defines the reference system. In our system we use a planar pattern for computing the intrinsic and extrinsic parameters of the camera stereo pair (Zhang, 2000). Using this approach, the coordinate system is placed in the world depending on the location of the calibration object. Therefore, joints' positions are referenced from an unknown world origin. To solve this problem, the coordinate system is automatically aligned with the user's position and orientation in the first frame. The reference system origin is placed in the virtual foot position of the user's model. Next, the y-axis is aligned to the unit vector that joins the user's foot and back and the x-axis is aligned to the unit vector that joins the user's right and left shoulder, setting the y component to zero.

Once the reference system is aligned with the user's position and orientation, 3D positions of the joints become environment independent because the origin

reference is aligned with the user's body and does not depend on the calibration process. However, the data still depends on the size of the user's limbs. A possibility to make data size invariant is given by the use of motion information of the joints through Euler angles (Moeslund et al., 2006b). Nevertheless, in this case, motion information is unstable, i.e., small changes of these values could give wrong detections. Alternatively, we propose a representation of each body limb by means of a unit vector, which represents the limb orientation. Formally, the unit vector that represents the orientation of limb, l, defined by joints J_1 and J_2, $\vec{u}_l$, is computed as follows

$$\vec{u}_l = \frac{\mathbf{J}_2 - \mathbf{J}_1}{\|\mathbf{J}_2 - \mathbf{J}_1\|}, \tag{4}$$

where $\mathbf{J}_i = (x_i, y_i, z_i)$ is the i-th joint 3D-position in the user's centered reference system. In this way, depending on the desired gesture alphabet, it is only necessary to compute the unit vector for the involved body limb. This representation causes data to be independent from the user's size and it solves spatial variations.

Once data is invariant, the next step is to represent a posture. We build the posture representation by using unit vectors of the limbs involved in the gesture set. The idea is to represent the user's body posture as a feature vector composed by all the unit vectors of the user's limbs. Formally, the representation of the orientation of a limb, l, is

$$\mathbf{q}^l = (u_x^+, u_x^-, u_y^+, u_y^-, u_z^+, u_z^-), \tag{5}$$

where u_x^+ and u_x^- are respectively the positive and negative magnitudes of the x-component of unit vector, u_x, note that $u_x = u_x^+ - u_x^-$ and $u_x^+, u_x^- \geq$

0. The same applies for components u_y and u_z. In this way, the orientation components of the limb unit vector are half-wave rectified into six non-negative channels. Therefore, by linking limb poses, we build the feature vector which represents the complete orientations of the user's limbs, see Eq. 6.

$$\mathbf{q} = \{\mathbf{q}^l\}_{l=1..n} \quad \sum_{l=1}^{n} \mathbf{q}^l = 1, \tag{6}$$

where n is the number of limbs involved in the motions to be recognized.

If we consider that a gesture is composed by several body postures, the motion feature vector is composed by the cumulative postures involved in its performance, that is

$$\hat{\mathbf{q}}_t = \frac{1}{T} \sum_{u=t-T}^{t} \mathbf{q}_u, \tag{7}$$

where T is its periodicity, and could be interpreted as a temporal window of cumulative postures. We state that this process encapsulates the temporal variations of gestures by means of detecting the periodicity of each user's motion performance in order to fix the T value, that is, its temporal extent.

Finally, for recognition, the key is to take advantage of the system's overall possibility of working in real-time. Therefore, before the recognition process starts it is possible to ask the user to perform several of the allowable motions in order to build a training set in real-time. Therefore, it is reasonable to assume that training motions near an unclassified motion should indicate the class of this motion. On the other hand, a motion is natural depending on the user's experience, as it has been shown in several experiments with children by Höysniemi et al. (2005). For this reason, we use the non-parametric technique of *k-nearest neighbors*. We employ a (k, v) nearest neighbor classifier that

Fig. 7. Interpretations of the *rotation* command by different users.

finds the k example motions closest to the current motion being performed by the user, and classifies this motion with the class that has the highest number of votes, as long as this class has more than v votes; otherwise the system considers that a significant motion has not been performed. Besides, we have tested how the users interpret each of the commands, mainly the complex commands, which are performed by users in completely different ways, see Fig. 7. This fact reinforces the selection of non-parametric techniques in order to make specific motion models easy for each user.

We measure similarity between the current motion, $\hat{\mathbf{q}}_t$, and the exemplars, $\hat{\mathbf{p}}_i$ with the Earth Mover's Distance (EMD), this is the measure of the amount of work necessary to transform one weighted point set into another. Moreover, it has been shown that bin-by-bin measures (e.g., L_p distance, normalized scalar product) are less robust than cross-bin measures (e.g., the Earth Mover's Distance (EMD), which allows features from different bins to be matched) for capturing perceptual dissimilarity between distributions (Rubner et al., 2000).

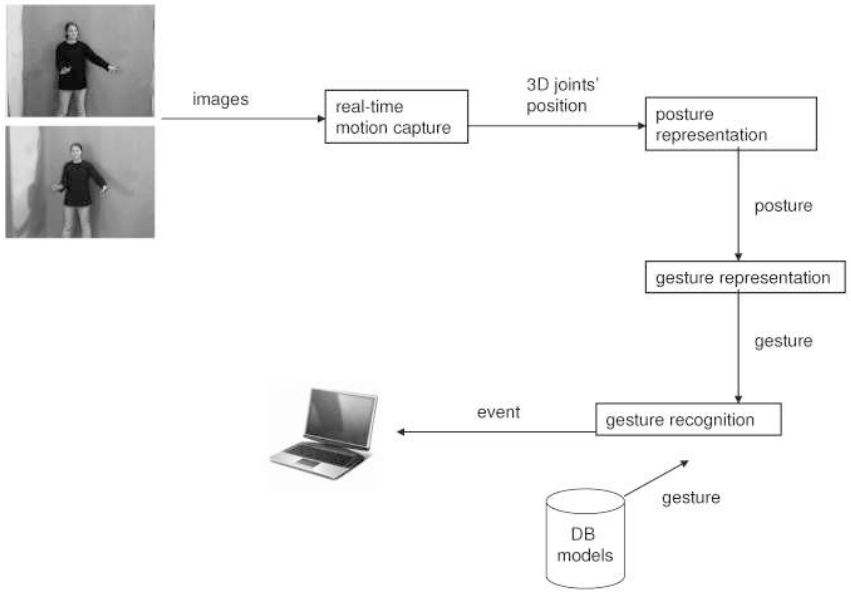

Fig. 8. Scheme of the gesture-based interface to generate the system's events.

4 A case study: videogame control through body gestures

In order to test our approach we have proposed to play a computer videogame interacting through user body gestures. Specifically, a free version of the classical Tetris videogame which has four different forms of control: *left*, *right*, *down* and *rotate*. Using the previously defined scheme of recognition, summarized in Fig. 8, it is possible to build an enactive interface that is flexible (taking into account the particular way of making gestures by each user), natural, intuitive and responsive to his actions.

The user is located in an interactive space that consists of a projection screen and is instrumented with a stereo camera pair. This configuration allows the user to view the videogame while performing its commands. Gestures occur in the workspace defined by the screens and the user.

The interface requirements are:

- Only one person shall be present in the space.
- The color of the users clothes should not be similar to skin color.
- The skin colored body parts, other than the hands and face, shall not be visible. For example, the user should not roll up his sleeves.
- In order to learn the motions in which user perform the commands, previously to starting the game, the system asks the user to make several isolated performances of each command. This is a way to automatically build the training set, i.e., the gesture models database.

The enactive interface was tested by three different users that had never experienced the application. We acquired three different sessions while producing all the necessary commands during the videogame. At this point, our dataset is formed by a training set composed of three performances for each command and for each different user, and a testing set with a total of 4500 frames containing different motions of each user playing the videogame. In addition, for comparison purposes, we also have conduced experiments using a Gaussian model to represent commands by computing the mean and variance of each user's learning motions. Results of both methods for a gesture periodicity of 10 frames, $T = 10$, are shown in Table 2.

The first interpretation of the results presented in Table 2 is that a user with no preparation can play this videogame in a natural way using only their own body motions (Fig. 9). In addition, the majority of errors are due to errors of the Vision-PIK estimation of the user's body joints. This fact implies that improving the system for capturing the user motions leads to better recognition performances.

Table 2

Gesture recognition results.

Method	Commands	Correct	Wrong	Non-Rec
Gaussian	86	72.09%	11.63%	16.28%
3-NN	86	86.05%	3.49%	10.46%

Fig. 9. Videogame control by recognizing the user motions in real-time.

5 Conclusions

Nowadays, there is a considerable effort in the research of human motion recognition methods due to its potential application in human-computer interaction. The majority of the previous works are based on using directly image values for recognition due to the difficulty of finding 3D body poses in real-time. Using image data implies the application of complex statistical models for recognition, which are difficult to use in practical applications.

We have presented an approach to obtain user's motion in a 3D-space. The main advantage of our system is that we avoid specifically invasive methods such as markers and that we allow the user to perform a broad range of motions. Moreover, the whole process is done in real-time to achieve a reliable interaction. By using an inverse kinematics based model, the system is potentially more accurate and robust to occlusion effects than approaches based on detection of pixel changes. This is because the model provides additional constraints that can be used to resolve any discrepancies between measured and predicted positions. We have tested the complete system with experiments to measure the accuracy of the end-effectors 3D tracking and the internal joint estimations and with an experiment where the users have to do several predefined motions. The quality of the results is sufficient for our objective, which is to open the way to exploit a non-invasive wide and coherent full-body postural space for real-time 3D interactions.

We have shown the potential of the system through an enactive interface. The novelty lies in the representation of pose that allows the interface to generalize over body shape differences in the population of users. Our approach is original

and it could be extended to represent more complex gestures and human activities. The complete system has been tested in a real-time application, a motion-based videogame control. The key idea is the use of a non-parametric technique, the k-nearest neighbor, for learning and recognition. Experiments have shown that, from a practical point of view, this technique of classification is appropriated for real world problems due to its simplicity in learning and on-line classification. Besides, the system adapts itself to each particular user's way of performing motions, avoiding a previous user's off-line training to learn the motions that can be recognized by the system.

As future work, this approach can be extended to more complex gestures than the ones shown in the presented application adding more limbs to the gesture representation. It is important to point out that our approach needs further testing. Specifically, it should be tested in real sessions with more users. These sessions should test how the number of learning exemplars affect the recognition of user's motions.

Acknowledgements

This work has been supported by EC grant IST-027110 for the HERMES project and by the Spanish M.E.C. under projects TIN2007-67993 and TIN2007-67896. J.Gonzàlez and J.Varona also acknowledge the support of a Juan de la Cierva and a Ramon y Cajal (cofunded by the European Social Fund) Post-doctoral fellowships from the Spanish MEC, respectively. Besides, the UIB team is a member of the Enactive Interest Group of the European Network of Excellence IST-002114-ENACTIVE Interfaces. Special thanks to Dr. Ronan Boulic from the Virtual Reality Lab of the Ecole Polytechnique Fédérale de

Lausanne for his invaluable help in this work and for providing the INKlib library that implements the Priority Inverse Kinematics algorithm.

References

Baerlocher, P., Boulic, R., 2004. An inverse kinematic architecture enforcing an arbitrary number of strict priority levels. Visual Computer 20 (6), 402–417.

Bobick, A. F., Davis, J. W., 2001. The recognition of human movement using temporal templates. IEEE Trans. Pattern Analysis and Machine Intelligence 23 (3), 257–267.

Bolt, R. A., 1980. 'put-that-there': Voice and gesture at the graphics interface. In: SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 262–270.

Bradski, G., Pisarevsky, V., 2000. Intels computer vision library. In: Proceedings of Computer Vision and Pattern Recognition (CVPR'00). Vol. 2. pp. 796–797.

Efros, A. A., Berg, A. C., Mori, G., Malik, J., 2003. Recognizing action at a distance. Proceedings of International Conference on Computer Vision (ICCV 2003).

Elgammal, A., Shet, V., Yacoob, Y., Davis, L., 2003. Learning dynamics for exemplar-based gesture recognition. In: Proceedings of Computer Vision and Pattern Recognition (CVPR'03). pp. 571–578.

Höysniemi, J., Hämäläinen, P., Turkki, L., Rouvi, T., 2005. Children's intuitive gestures in vision-based action games. Commun. ACM 48 (1), 44–50.

InterSense Inc. Website, http://www.isense.com (Last accessed June 2008).

Kojima, A., Izumi, M., Tamura, T., Fukunaga, K., 2000. Generating natural language description of human behavior from video images. In: Proceedings of 15th International Conference on Pattern Recognition. Vol. 4. pp. 4728–4731.

Moeslund, T. B., Hilton, A., Krger, V., 2006a. A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding 104 (2–3), 90–126.

Moeslund, T. B., Reng, L., Granum, E., 2006b. Finding motion primitives in human body gestures. In: Gesture in Human-Computer Interaction and Simulation: 6th International Gesture Workshop. pp. 133 – 144.

O'Hagan, R. G., Zelinsky, A., Rougeaux, S., 2002. Visual gesture interfaces for virtual environments. Interacting with Computers 14 (3), 231–250.

Rao, C., Yilmaz, A., Shah, M., 2002. View-invariant representation and recognition of actions. International Journal of Computer Vision 50 (2), 203–226.

Ren, H., Xu, G., Kee, S., May 2004. Subject-independent natural action recognition. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition. pp. 523–528.

Rubner, Y., Tomasi, C., Guibas, L., 2000. The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40 (2), 99–121.

Trucco, E., Verri, A., 1998. Introductory Techniques for 3D Computer Vision. Prentice-Hall.

Turk, M., Kolsch, M., 2004. Emerging Topics in Computer Vision. Prentice Hall, Ch. Perceptual interfaces.

Varona, J., Buades, J., Perales, F., 2005. Hands and face tracking for vr applications. Computers and Graphics 29 (2), 179–187.

Wu, Y., Huang, T. S., 1999. Vision-based gesture recognition: A review. In:

Gesture Workshop'99, LNAI 1739. pp. 103–115.

Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (11), 1330–1334.