

# Unraveling Natural Interfaces for Co-Located Groupware: Lessons Learned in an Experiment with Music

João Carreira and Paulo Peixoto

Institute of Systems and Robotics, University of Coimbra, Portugal

Email: {joaoluis, peixoto}@isr.uc.pt

**Abstract**— The computer is an ubiquitous element of modern society, nonetheless, human computer interaction is still rather inflexible. Particularly in local collaborative environments, like office meetings, the property that the mouse and keyboard exhibit of being a gateway for the individual to act upon a workspace, has barred the way to the production of co-located collaboration technologies, because the users have to time-share their actions upon the workspace. Despite recent developments in touch sensitive multi-user tabletops, we still believe the portability, low cost and potentially large input area of vision sensors presents the most promising approach to unravel natural human computer interfaces.

We present in this paper our design of a computationally inexpensive vision based interface that allows multiple users to interact simultaneously with a single computer by performing hand gestures, which are filmed by a static video camera. This interface attempts to continuously recognize predefined postures and movements using a view-dependent method. We also present A.C.O, a co-located groupware application that receives input from the vision-based interface and allows users around a table to collaborate playing synthesized music instruments by moving their hands. This prototype gave us important hints on the more immediate obstacles the technology must overcome.

**Index Terms**— co-located groupware, human computer interface, gesture recognition, computer vision

## I. INTRODUCTION

Remote groupware, software that is designed to be used by multiple users, has been on the rise since the advent of high speed networks and the internet – with the Wikipedia and online gaming being good examples of it. The same can't be said of synchronous co-located groupware - software used simultaneously by multiple users sharing an input device - because the mouse and keyboard are notoriously disruptive to co-located collaboration in the sense that they require users to

interrupt each other, in order to time-share access to a single computer.

One can argue that computer vision is the most promising technology for the creation of a natural human computer interface (HCI). Some factors behind this notion are the ever-shrinking size of digital devices which demands for "device-external interfaces" [1], the need for large, low cost input areas where multiple users can interact and collaborate mediated by a computer, and the unobtrusiveness of vision sensors at capturing hand gestures – arguably the most natural mean for a human to communicate. Pavlovic [2] noted however that, ideally, this naturalness requires that any and every gesture performed by the user should be interpretable, but that the state of the art in vision-based gesture recognition is far from providing a satisfactory solution to this problem. It still is, but there have been solid improvements in both theory and practice.

This paper provides some new insights, more detail, and a slight reframing of the research originally published in [19]. Its primary contribution is a report on an empirical study about the requirements of an interface for co-located groupware, and a diagnostic of specific problems the computer vision community will need to overcome in order to make this kind of application mainstream, and, ultimately, further empower computer users.

## II. COMPARISON WITH PREVIOUS WORK

In traditional HCI, most of the proposed systems have used some device, such as an instrumented glove, for incorporating gestures into the interface. If the goal is natural interaction in everyday situations this might not be acceptable. However, a growing number of applications of hand gesture recognition for HCI exist. Mostly they require restricted backgrounds and camera positions, and a small set of gestures, performed with one hand. They can be classified as applications for pointing, presenting, digital desktops, virtual workbenches and VR. Digital desktops are applications that aim at developing mixed reality desktops, using free hand pointing and manipulation of digital objects. Krueger's VideoDesk [3]

---

Based on "A Vision Based Interface for Local Collaborative Music Synthesis", by J. Carreira, and P. Peixoto which appeared in the Proceedings of the IEEE International Conference on Face and Gesture Recognition 2006, Southampton, UK, April 2006. © 2006 IEEE.

was an early desk-based system in which an overhead camera and a horizontal light was used to provide hand gesture input for interactions, which were then displayed on a monitor at the far end of the desk. Maggioni and K ammerer [4] explored pointing gestures in vision-based virtual touch screens for office applications, public information terminals and medical applications. The detection is based on a skin segmentation step, and the approach requires controlled backgrounds. More recently, Koike et al. [5] developed an augmented desk interface, EnhancedDesk, with computer vision as a key technology. EnhancedDesk uses a projector for presenting information onto a physical desktop, an infrared camera for detecting users' arms, hands, and hand poses, and a pan-tilt camera for giving detail. Users can manipulate digital information directly by using their hands and fingers.

Recently some touch based multi-user interfaces we're unveiled, as the MERL's DiamondTouch [18]. This particular device is much more robust than current vision based technology. On the other hand it is very expensive, not portable, and can only directly capture information about the shape of the pressing areas of the hand against the table.

In this paper we propose a fast vision based interface (VBI) that we designed, which is on par with state of the art VBI's: capable of detecting and tracking multiple bare hands under arbitrary poses and of recognizing a set of predefined gestures. As in most related research, our experimental configuration consists of a video camera connected to a computer, pointing down at a table in order to capture the hand gestures of users standing around it, in a typical collaboration environment.

For the last 20 years the 2-D desktop paradigm as the user's interface has ruled. We are now focusing on augmenting this 2-D paradigm to a real table, with a horizontal LCD display on top. People would work with the system using their hands over the table and acting upon the objects in the image. We believe this to be more natural if the hands are as near as possible to the display. Also, we observed, and has been reported in the literature [21] that using hand gestures for interfacing with the computer, without physical support for the arms, can cause discomfort after some time. Summing it up, we believe that the most frequently done gestures in a tabletop scenario will happen in a plane parallel to the image plane, and that's the type of gestures our system works with.

We also present a prototypical groupware application: A.C.O, an interactive music synthesis environment which relies on the proposed VBI, and where several users can collaborate, controlling musical instruments, by performing gestures with their hands.

We believe the experience gathered using this approach holds some helpful domain knowledge for researchers.

### III. OVERVIEW OF THE INTERFACE

We aimed at creating a rich VBI, which would be able to retrieve the positions and orientations of the users' hands in the image, and to recognize a predefined set of seven postures and four movements, independently of the users location around the table. It would also be important for the VBI not to be computationally too demanding, in order to be usable with the off the shelf hardware most computers have. The dataflow in the VBI is outlined in Fig. 1, and consists of a cascade of modules that process every frame captured by the camera. The tracking and recognition modules keep state, while the detection and segmentation modules are purely reactive.

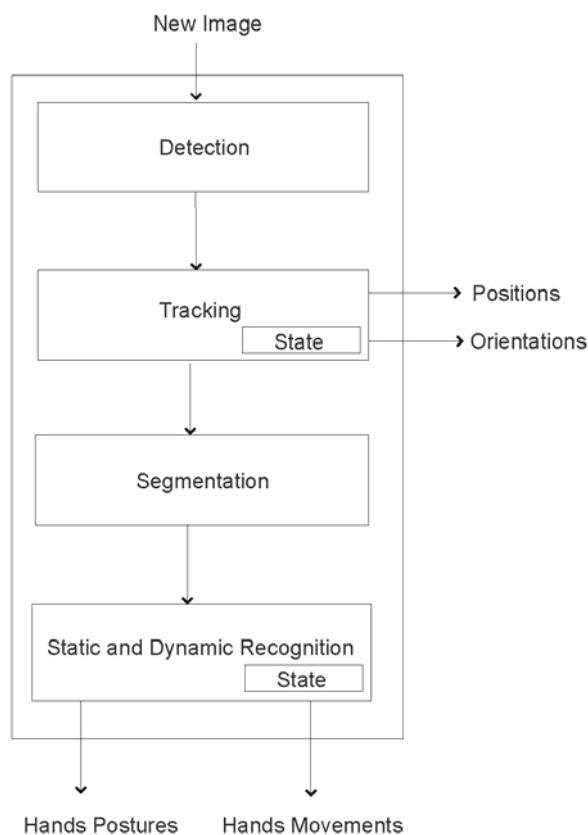


Figure 1. The VBI's dataflow

We employ the simple histogram retroprojection method for skin color recognition in HSV space presented in [6], which purpose is threefold: it's used in combination with a Haar-Like Features Object Detector [8] using the extend feature set presented in [9] trained with a single hand posture for hand detection; its output is fed to CAMSHIFT [7][15] which tracks each hand's position in the image from frame to frame; and is thresholded and treated for segmentation of the hand from the background, to serve as input to a posture recognition module. The result of this process is illustrated in Fig. 2.

The system is highly dependent on the quality of this process, as errors in it will propagate to all modules. This process is also responsible for the major limitations of the interface: the requirements for the absence of skin like

colored objects in the table, and for the users to wear long sleeves to separate the hands from the forearms. In the literature, these problems have previously been solved using color markers, or some heuristic for cutting the blob in the wrist zone. Our **first lesson** was that **skin color cues don't seem informative and robust enough by itself, for tracking and segmenting the hands.**

The purpose of the detection module is to initialize the process of tracking the position of a hand in the image. After a hand is detected, it is tracked between frames, enabling the recognition of spatio-temporal movements. CAMSHIFT is a popular tracking algorithm, which is very fast and handles deformable objects well, but can't handle occlusions by similar objects, so we assume that there are no occlusions between the hands. Each tracked hand is segmented from the background and the resulting blob's contour is processed, and classified by a Support Vector Machine [13] trained with a set of seven postures, represented in Fig. 3. The tracked hand's position is appended to the sequence of past positions, which is continuously preprocessed and classified using Hidden Markov Models (HMMs) of a set of movements, which are also represented in Fig. 4.

In order to train the Object Detector, we collected around 400 positive training samples of left open hands with the palm directed at the camera, and 4000 negative training samples. The training samples size chosen was 20x20 pixels, and the detected hand posture was one we feel is the most natural - an open hand with the palm up is historically used for asking for something, in this case for being detected. Notwithstanding, it's a difficult posture to be learned through Adaboost by the Object Detector [10] due to the concavities between the fingers, which introduce noise. We trained a cascade with 15 stages, which aggregated a total of 147 weak classifiers, for a theoretical hit rate of 93% and false alarm rate of 0.003%. Its real performance was not that good, possibly because of a disproportionate ratio of positive to negative training samples, and the difficulty presented by the detected posture. In order to minimize this problem, we passed the detected objects through an acceptance skin color recognition test. If the amount of skin colored pixels inside the rectangle indicated by the Object Detector surpasses a threshold, the object is validated as a hand.

We perform hand detection on the acquired images, on its horizontally and vertically flipped versions, to detect right hands and hands of people that are in the opposite side of the table, respectively.



Figure 2. The image on the right is the result of the application of the skin color recognition scheme to the image on the left. Darker pixels correspond to lower probabilities of belonging to skin.

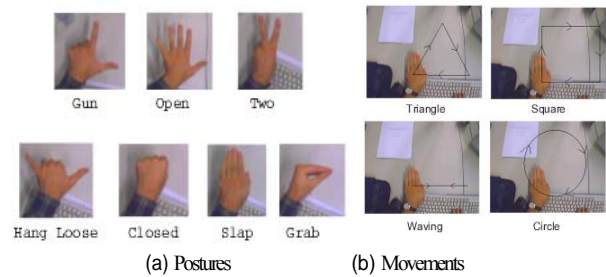


Figure 3. Recognized postures and movements.



Figure 4. Sample gray level images used to train the object detector.

While we can detect hands, we can't *undetected* them: the detector provides no evidence on the absence of hands in the image. Only if our detector was aware of all possible hand configurations – which may not be reasonable given the wide range of hand appearances - would the absence of detection indicate that there was no hand present. As we only detect a single appearance, we have no *undetected* capability (we actually use the area of skin color being tracked as an heuristic to *undetected* hands). **Lesson number two is: create an über-detector or a detector and an undetector.**

#### IV. POSTURE AND MOVEMENT RECOGNITION

In this section we describe the methods we used to recognize the set of predefined gestures. All of them were made by a single hand, and were either simple postures or simple movements. As these modalities are orthogonal to each other, the VBI attempted to recognize both in parallel during operation.

##### A. Posture Recognition

In order for the VBI to be insensitive to the position of the users around the table, it was important for the posture recognition module to be invariant to rotation. Also, for the system to be usable by different people, with different hand sizes, and for adaptability to different distances from the hand plane to the camera, the system would desirably provide scale invariance. To allow users to use both hands, the system should also be invariant to symmetry. Finally, as postures are independent of the position of the hands in the image, it should also present translation invariance.

These requirements were greatly eased by the hand's shape representation chosen, with the exception of the invariance to symmetry. To compute this representation we consider the magnitude part of the polar coordinates of each point of the hand's contour in relation to its center of mass, divide the contour into fixed radial segments and for each segment we choose the maximum amplitude of any point there, creating the vector

$$\mathbf{x} = [x_1 \dots x_n]^t.$$

Then we calculate the index  $m$  whose corresponding magnitude  $r_m$  is the maximum, and construct a new vector as

$$\mathbf{x}' = [x_m/x_m \ x_{m+1}/x_m \dots \ x_1/x_m \dots \ x_{m-1}/x_m]^t.$$

Since the number of points in a contour is variable along time it was important to uniformize the length of the contour signature. The proper number of radial segments to consider depends on the typical distance of the camera from the table and on its intrinsic parameters, in our case 80 points were enough to capture the most nuanced posture - "Open", which also has the biggest contour because of the concavities between the fingers.

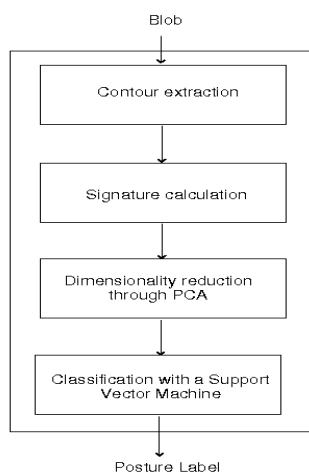


Figure 5. The posture recognition process employed

As we used a referential centered in the hand, translation invariance was guaranteed. To achieve scale invariance we divided the magnitudes of all the points in the signature by the largest one. The fact that the first dimension of the signature corresponds to the point of the hand's contour farthest away from its center provides invariance to rotation as long as this point corresponds to the same part of the hand. In the case of postures which present points whose magnitude approximates that of the largest one, the class conditional probabilities become multimodal - a difficulty we relied on the classifier to handle. In order to classify equally both left and right hand postures, we calculated from one posture's signature the signature of the symmetric posture, and tried to classify both. Then we chose the most probable. Let the signature of the posture be

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^t.$$

Then its symmetric is, as follows from the signature calculation process,

$$\mathbf{x} = [x_1 \ x_n \ x_{n-1} \ \dots \ x_2]^t.$$

There is much redundant information in these signatures. Principal Component Analysis (PCA) was used to eliminate linear correlations and reduce the dimensionality of the data while maintaining most of the relevant information [14]. Having a signature vector with higher signal-to-noise ratio and reduced dimensionality allows significant savings in computational power and reduces the complexity of the classification process.

Let  $\mathbf{x}$  denote a  $d$ -dimensional signature vector, and  $A$  a matrix composed by the first  $N$  eigenvectors of the covariance matrix  $P$  of all the data collected, disposed by columns and ordered by decreasing value of their corresponding eigenvalues, and  $\mu$  the mean of the data. The projection  $\mathbf{x}'$  of  $\mathbf{x}$  into the new eigenspace is given by:

$$\mathbf{x}' = A^t(\mathbf{x} - \mu)$$

and its reconstruction by:

$$\mathbf{x} = A\mathbf{x}' + \mu$$

The fraction  $E(N)$  of the total variance of the data accounted by the first  $N$  principal components is given by:

$$E(N) = \frac{\sum_{i=1}^N \lambda_i}{\sum_{i=1}^d \lambda_i}$$

One of the decisions that needed to be made concerning PCA was the number of principal components to apply. Our approach was to try to classify ourselves the reconstructions of signatures, projected to eigenspaces formed by a different number of principal components, and to make a judgment about which eigenspace allowed for a minimally comfortable discrimination and recognition of all the gestures. We concluded that 20 principal components were appropriate, which accounts for 83.1 % of the variance present in the data. Fig. 6 illustrates this analysis.

We gathered 1500 80-dimensional signature vectors for each posture, giving a total of 10500 samples, all of them performed by the same user, using its left hand with its back directed at the camera, as shown in Fig. 3. Then we converted this dataset to the PCA space, and trained a Support Vector Machine using 4-fold cross-validation. After some experiments - whose results are presented in Table I - we decided to use a RBF kernel and set  $C = 1$ , which resulted in a very high rate of correct classifications during cross-validation: about 99.3 %.

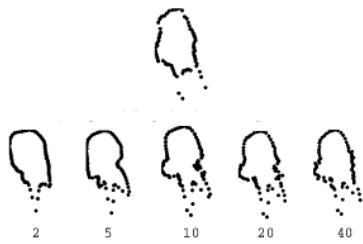


Figure 6. Signature reconstruction of the posture "Two" using the first N principal components.

We can now draw **lesson three: given a good segmentation of the hand's shape, modern classifiers and a simple representation will do the job.**

### B. Movement Recognition

We wanted the VBI to continuously try to recognize movements, and to do it with invariance to translation of the hands and slight temporal and spatial scaling. We relied on the features chosen to achieve the first property and on the nature of Hidden Markov Models for the second. The feature vector chosen,  $\mathbf{v} = [v_x \ v_y]^t$ , was simply the difference of the hand's position  $\mathbf{p}$  in image coordinates, between consecutive images:

$$\mathbf{v}(t) = \mathbf{p}(t) - \mathbf{p}(t-1)$$

It was verified that the camera frame grabbing rate was approximately constant, and so there was no need to divide  $\mathbf{v}(t)$  by the period of time between frames.

We trained a HMM for each predefined movement, adopting a Bakis [20] topology and modeling the observation probability distributions as mixtures of laplacians. We further created a set of simple models of translations to the right, left, up and down, so as to increase the specificity of the target movements. The models were trained with the Segmental K-Means algorithm [11], which we applied on a dataset containing 20 samples of each movement - much smaller than the one used for training the posture classifier because it was observed to take much more time to collect - and for the continuous recognition we used the Viterbi algorithm [12]. Due to the small dataset's size, we opted to measure the performance of the classifier online, by having a human expert observing a user of the VBI and classifying his movements according to his personal judgment - which we considered the ground truth - and by comparing the expert's and the recognition module's answers. The results are presented in Table II. The movements recognized aren't ambitious enough for learning much of a lesson.

## V. THE MUSIC SYNTHESIS ENVIRONMENT

As a demonstration we developed a simple software application named A.C.O - as "Another Camera Orchestra" - that receives its sole input from the presented VBI, in order to evaluate its performance.

A.C.O allows multiple users to play synthesized musical instruments simultaneously, featuring the possibility of different users control different parameters of the same instrument in real time. We used the Synthesis Toolkit (STK) [17] to synthesize the sounds.

TABLE I.

FRACTION OF CORRECT CLASSIFICATIONS USING 4-FOLD CROSS-VALIDATION.

Polynomial Kernel	C=0.01	C=0.1	C=1
Open	0.976	0.99	0.998
Two	0.992	0.998	1
Grab	0.99	0.998	1
Gun	0.984	0.99	1
Hangloose	0.986	0.996	1
Slap	0.98	0.986	0.998
Closed	1	1	1
Cross-Validation	0.984	0.989	0.998
RBF Kernel	C=0.01	C=0.1	C=1
Open	0.954	0.99	0.998
Two	0.99	0.996	1
Grab	0.89	0.996	1
Gun	0.984	0.994	1
Hangloose	0.98	0.994	1
Slap	0.964	0.984	0.998
Closed	1	1	1
Cross-Validation	0.951	0.989	0.993

TABLE II.

HMM CLASSIFIER PERFORMANCE, DURING A SESSION OF 100 MOVEMENTS PERFORMED BY A SINGLE USER.

Movement	Fraction of correct classifications
Circle	0.85
Square	0.75
Triangle	0.80
Waving	1.00

This application maps each posture to a different instrument - i.e. "Slap" corresponds to drums, "Two" to an organ, "Gun" to a guitar and "Hang loose" to a cither. The tones are determined by the horizontal coordinates of the hands in the image, and consequently in the table, and their triggering is accomplished by variations in the vertical coordinates of the hands. For users to effectively collaborate playing the same instrument, we implemented a gestural protocol in which groups get formed through a rendezvous characterized by the execution of the same movement, nearly at the same time, by more than one hand. A group is broken when one of the hands does the "Waving" movement. When a group is formed, the hand that started the rendezvous controls which instrument to play, the tones, and triggers the synthesis of sounds, while other hands can control volume, reverb or pitch.

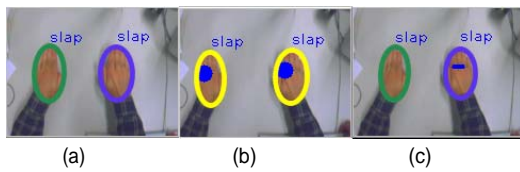


Figure 7. A.C.O in action with two hands in the image. In a) there is no group, in b) both hands performed the “Circle” movement and a group was formed, in c) the group has been broken by a “Waving” movement.

The application was implemented in a Pentium 4 2.4 Ghz machine, and was able to process input images from the scene at the full frame rate provided by the firewire camera used - 15 frames per second.

In spite of its simplicity, A.C.O did expose some unpredicted limitations of the interface. In fact, users reported difficulty concerning the ability to control the rhythm of the instruments - controlling how long each note plays. Also, playing a melody precisely was reported as being a difficult task. Being in an early stage of the investigation, we have yet to perform a formal usability study. Nevertheless, we suspect these problems share a common cause, which is, in general terms, the process of triggering an action. We concluded from the experiments that this was related to the complete separation between movement and posture classification. That was **lesson number four: a classifier that can recognize slight posture alterations should enable to user to command precisely the “when” of an action. Classify these kinds of movements jointly with posture.**

Other than that, some users told us that using this application was fun, and we found that they learned to collaborate and play the instruments very quickly.

## VI. CONCLUDING REMARKS

The lack of suitable alternatives to traditional input devices as the mouse and the keyboard has hindered the development of rich computer mediated local collaboration experiences. Vision based interfaces capable of recognizing hand gestures are a promising technology, as they allow users to interact freely with the computer without the need for any special external device.

This paper first described one such vision-based interface, capable of detecting and tracking multiple users' hands and of accurately recognizing a predefined set of postures and movements. This interface uses a view-dependent posture recognition method based on the classification of contour signatures using a Support Vector Machine, and Hidden Markov Models for continuous recognition of movements. The posture recognition method is invariant to rotation, scaling and symmetry, but is dependent on the use of non-skin colored sleeves to separate the hands from the forearms. The area captured in the image primarily limits the number of hands being simultaneously tracked.

Due to the configuration of the interface, it's hard to tell if any two hands in the table belong to a single person. While this grouping can be achieved if the user

makes the *rendez-vous* procedure with both hands, in applications where two-handed gestures would be frequent, it could be desirable that this happened automatically. One possibility would be to try to infer if any two hands belongs to the same user by analyzing their visual similarity and maybe other factors such as proximity, and the appearance of the arm, but that hasn't been explored in this work.

Secondly, we presented A.C.O, a rudimentary collaborative music synthesis application, which relies on the developed vision-based interface to recognize and map gestures performed by multiple users standing around a table, to sounds of several musical instruments. Although there are plenty of paths to explore beyond the simple one to one mapping between postures and instruments of our approach, the main goal of this application was to serve as a test bench on the expressive needs of vision based interfaces.

We learned four lessons, which we now restate:

- Skin color cues don't seem informative and robust enough by itself, for tracking and segmenting the hands.
- You should use a hand's *über*-detector or a detector and an *undetector*.
- Given a good segmentation of the hand's shape, modern classifiers and a simple representation will classify it correctly.
- A classifier that can recognize slight posture alterations should enable users to command precisely the “when” of an action. Classify these kind of movements jointly with posture.

As future work we envision tackling the identified problems, as well as facing some new ones when we augment the table with visual feedback provided by a video projector or a large display.

We're also planning on exploring gesture recognition in 3 dimensions. Being able to recognize gestures where depth is important would break with the conventional desktop paradigm, and would certainly generate hype and enlarge the potential of computer users. We can easily imagine applications for gestures where depth could be important: e.g. zooming a document by assuming a "grab" hand posture and then pulling the hand up. A straightforward way to add depth sensitivity to our system is to use multiple cameras. In that case we can compute disparities and have depth information for some locations in the image. We would leave the posture recognition part untouched, and simply add the depth parameters to the movement features. The depth information would be mostly directly used by the applications (in the zoom example above, the document processing application would use the depth information to set the level of zoom).

## ACKNOWLEDGMENTS

Research described in the paper was financially supported by FCT under grant No. POSC/EEA-SRI/61451/2004. The first author was financially supported by FCT PhD grant SFRH/BD/24295/2005

## REFERENCES

- [1] M. Kölsch, "Vision Based Hand Gesture Interfaces for Wearable Computing and Virtual Environments", Phd thesis, 2004.
- [2] V.I Pavlovic, R. Sharma, and T.S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review". In IEEE Trans. on PAMI. 19 (7) 677-695, 1997.
- [3] M. Krueger, "Artificial Reality II". Addison-Wesley, 1991.
- [4] C. Maggioni and B. Kämmerer, "Gesture Computer - History, Design and Applications". In Cipolla and Pentland (eds) Computer Vision for Human-Computer Interaction, Cambridge University Press, pp. 23-51, 1998.
- [5] H. Koike, Y. Sato, and Y. Kobayashi, "Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System". ACM ToCHI, Vol 8, no. 4, pp.307-322, 2001.
- [6] M. J. Swain and D. H. Ballard, "Color Indexing". Intl. Journal of Computer Vision, pp. 11-32, 1991
- [7] G. R. Bradski, "Real-time face and object tracking as a component of a perceptual user interface". Proc. IEEE Workshop on Applications of Computer Vision, pages 214-219, 1998.RL
- [8] P. Viola and M. Jones, "Robust Real-time Object Detection". In International Workshop on Statistical and Computational Theories of Vision, July 2001.
- [9] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection". Proc. IEEE Intl. Conference on Image Processing, volume 1, pages 900-903, Sep. 2002.
- [10] M. Kölsch and M. Turk, "Robust Hand Detection". Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition May 2004.
- [11] B. Juang and L.R. Rabiner, "The Segmental K-Means Algorithm for Estimating Parameters of Hidden Markov Models". Transactions on acoustics, speech and signal processing, vol. 38, no 9, pp. 1639-1641, 1990.
- [12] L.R. Rabiner, "A Tutorial on Hidden {Markov} Models and Selected Applications in Speech Recognition". Proc. IEEE ,vol. 77, pp. 257-285, 1989.
- [13] Nello Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods". Cambridge University Press, 2000.
- [14] R. Duda, P. Hart, G. Stork, "Pattern Classification". Wiley, 2001
- [15] Intel Corporation Open Source Computer Vision Library. Url: <http://www.intel.com/research/mrl/research/opencv>
- [16] Lti-Lib Library, Url: <http://ltilib.sourceforge.net/doc/homepage/index.shtm>
- [17] Perry R. Cook & Gary P. Scavone, Synthesis Toolkit (STK). Url: <http://ccrma.stanford.edu/software/stk>
- [18] Deitz, P. & Leigh, D. (2001). DiamondTouch: A Multi-User Touch Technology. In Proceedings of UIST'00: ACM Symposium on User Interface Software and Technology, 5 8 November 2000, San Diego, CA, pp. 219-226.
- [19] J. Carreira, P. Peixoto, "A Vision Based Interface for Local Collaborative Music Synthesis". Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, April 2006.
- [20] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang, "A model-based connected-digit recognition system using either hidden Markov models or templates," *Comput. Speech Lang.*, pp. 167-197, 1986.
- [21] D. M. Krum, O. Omoteso, W. Ribarsky, T. Starner, and L. F. Hodges. Evaluation of a Multimodal Interface for 3D Terrain Visualization. In IEEE Visualization, pages 411-418, October 27-November 1 2002.

**João Carreira** received his B.Sc. degree in Electrotechnical and Computer Engineering from University of Coimbra in 2005.

He was also an intern at the Portuguese company Critical Software, programming a prototype of a satellite data handling system, and a programmer in a joint project between the Institute of Systems and Robotics and the Spanish company Visual Tools, developing an intelligent surveillance system. He is currently a PhD student at the Department of Electrotechnical and Computer Engineering of the University of Coimbra and his research interests are computer vision with a strong focus on machine learning techniques, and also human-computer interfaces using gesture recognition.

**Paulo Peixoto** received his B.Sc. degree in Electrical Engineering and M.S. degree in Systems and Automation from the University of Coimbra in 1989 and 1995, respectively. In 2003 he received a PhD in Electrical Engineering from University of Coimbra.

He is a member of the Institute of Systems and Robotics (ISR), where he is a researcher. His research interests include computer vision, visual surveillance and human computer interaction.