Hand Gesture Recognition for Human-Machine Interaction.

Conference Paper · January 2004

Source: DBLP

Elena Nielsen
Universidad de La Laguna

Hand Gesture Recognition for Human-Machine Interaction

Elena Sánchez-Nielsen
Department of Statistic,
O.R. and Computer Science,
University of La Laguna
Edificio de Física y Matemáticas
38271, La Laguna, Spain
enielsen@ull.es

Luis Antón-Canalís
Department of Statistic,
O.R. and Computer Science
University of La Laguna
Edificio de Física y Matemáticas
38271, La Laguna, Spain
woodwolf@telefonica.net

Mario Hernández-Tejera
Institute of Intelligent Systems and
Numerical Applications in
Engineering.
Campus Universitario de Tafira
35017, Las Palmas, Spain
mhernandez@dis.ulpgc.es

ABSTRACT

Even after more than two decades of input devices development, many people still find the interaction with computers an uncomfortable experience. Efforts should be made to adapt computers to our natural means of communication: speech and body language. The PUI paradigm has emerged as a post-WIMP interface paradigm in order to cover these preferences. The aim of this paper is the proposal of a real time vision system for its application within visual interaction environments through hand gesture recognition, using general-purpose hardware and low cost sensors, like a simple personal computer and an USB web cam, so any user could make use of it in his office or home. The basis of our approach is a fast segmentation process to obtain the moving hand from the whole image, which is able to deal with a large number of hand shapes against different backgrounds and lighting conditions, and a recognition process that identifies the hand posture from the temporal sequence of segmented hands. The most important part of the recognition process is a robust shape comparison carried out through a Hausdorff distance approach, which operates on edge maps. The use of a visual memory allows the system to handle variations within a gesture and speed up the recognition process through the storage of different variables related to each gesture. This paper includes experimental evaluations of the recognition process of 26 hand postures and it discusses the results. Experiments show that the system can achieve a 90% recognition average rate and is suitable for real-time applications.

Keywords

Man-Machine Interaction, Perceptual user interface, Image Processing, Hand gesture recognition, Hausdorff distance

1. INTRODUCTION

Body language is an important way of communication among humans, adding emphasis to voice messages or even being a complete message by itself. Thus, automatic posture recognition systems could be used for improving human-machine interaction [Turk98]. This kind of human-machine interfaces would allow a human user to control remotely through hand postures a wide variety of devices. Different applications have been suggested, such as the contact-less control or home appliances for welfare improvement [Pen98, Wil95, Ju97, Dam97].

Journal of WSCG, Vol.12, No.1-3, ISSN 1213-6972
WSCG'2004, February 2-6, 2003, Plzen, Czech Republic.

In order to be able to represent a serious alternative to conventional input devices like keyboards and mice, applications based on computer vision like those mentioned above should be able to work successfully under uncontrolled light conditions, no matter what kind of background the user stands in front of. In addition, deformable and articulated objects like hands mean an increased difficulty not only in the segmentation process but also in the shape recognition stage.

Most work in this research field tries to elude the problem by using markers, using marked gloves, or requiring a simple background [Dav94, Bob95, Hun95, Mag95]. Other approaches are based on complex representations of hand shapes, what makes them unavailable for their implementation in real-time applications [Tri01].

A new vision-based framework is presented in this paper, which allows the users to interact with computers through hand postures, being the system adaptable to different light conditions and backgrounds. Its efficiency makes it suitable for real-time applications. The present paper focuses on the diverse stages involved in hand posture recognition, from the original captured image to its final classification. Frames from video sequences are processed and analyzed in order to remove noise, find skin tones and label every object pixel. Once the hand has been segmented it is identified as a certain posture or discarded, if it does not belong to the visual memory.

The recognition problem is approached through a matching process in which the segmented hand is compared with all the postures in the system's memory using the Hausdorff distance. The system's visual memory stores all the recognizable postures, their distance transform, their edge map and morphologic information. A faster and more robust comparison is performed thanks to this data, properly classifying postures, even those which are similar, saving valuable time needed for real time processing. The postures included in the visual memory may be initialized by the human user, learned or trained from previous tracking hand motion [San99] or they can be generated during the recognition process.

This paper is organized as follows: Section 2 introduces an overview of system components. Hand gesture detection and recognition approach are presented in Section 3 and 4. The advantages of the proposed system are demonstrated on experimental evaluations from real world scenes in Section 5. Conclusions and future work are finally described in Section 6.

2. SYSTEM COMPONENTS

A low cost computer vision system that can be executed in a common PC equipped with an USB web cam is one of the main objectives of our approach. The system should be able to work under different degrees of scene background complexity and illumination conditions, which shouldn't change during the execution.

Figure 1 shows an overview of our hand posture detection and recognition framework, which contains two major modules: (i) user hand posture location; and (ii) user hand posture recognition. The following processes compose the general framework:

1. **Initialization**: the recognizable postures are stored in a visual memory which is created in a start-up step. In order to configure this memory, different ways are proposed.

2. **Acquisition**: a frame from the webcam is captured.

3. **Segmentation**: each frame is processed separately before its analysis: the image is smoothed, skin pixels are labelled, noise is removed and small gaps are filled. Image edges are found, and finally, after a blob analysis, the blob which represents the user's hand is segmented. A new image is created which contains the portion of the original one where the user's hand was placed.

4. **Pattern Recognition**: once the user's hand has been segmented, its posture is compared with those stored in the system's visual memory (VMS) using innovative Hausdorff matching approach.

5. **Executing Action**: finally, the system carries out the corresponding action according to the recognized hand posture.

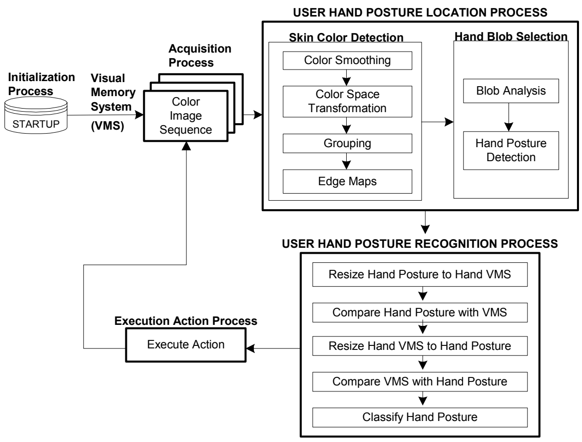


Figure 1. Global Hand Posture Detection and Recognition Diagram.

3. HAND POSTURE DETECTION

The operators developed for image processing must be kept low time consuming in order to obtain the fast processing rate needed to achieve real time speed. Furthermore, certain operators should be adaptable to different light conditions and backgrounds.

Skin Colour Features

Modelling skin colour requires the selection of an appropriate colour space and identifying the cluster associated with skin colour in this space. HSI space (Hue, Saturation and Intensity) was chosen since the hue and saturation pair of skin-tone colours are independent of the intensity component [Jon98]. Thus, colours can be specified using just two parameters instead of the three specified by RGB space colour (Red, Green, Blue).

In order to find common skin tone features, several images involving people with different backgrounds and light conditions where processed by hand to separate skin areas. Figure 2 shows the distribution of skin colour features. Yellow dots represent samples of skin-tone colour from segmented images, while blue dots are the rest of the image colour samples. It can be observed that skin-tone pixels are concentrated in a parametrical elliptical model. For practical purposes, however, skin-tone pixel classification was simplified using a rectangular model. The fact that the appearance of the skin colour tone depends on the lighting conditions was confirmed in the analysis of these images. The values lay between 0 and 35 for hue component and between 20 and 220 for saturation component.

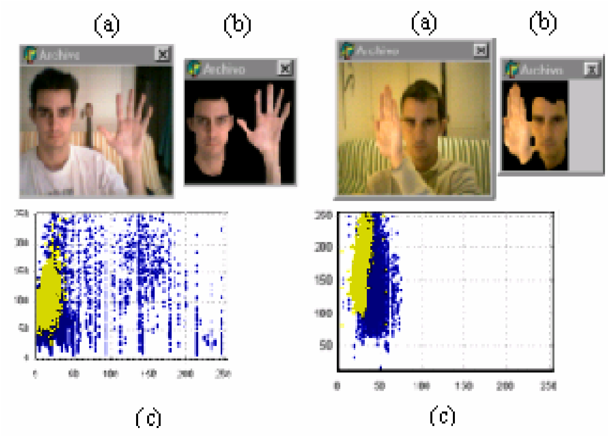


Figure 2. Skin-tone colour distribution in HSI space: (a) original image under natural and artificial light conditions; (b) segmented- skin image; (c) HSI colour space (yellow dots represent skin colour samples and blue dots represent the rest of the image samples), *x-axis* for hue component and *y-axis* for saturation component.

Colour Smoothing

An image acquired by a low cost web cam is corrupted by random variations in intensity and illumination. A linear smoothing filter was applied in order to remove noisy pixels and homogenize colours. Best results were achieved using a mean filter, among the different approaches of proposed lineal filters [Jai95].

The appearance of the skin-tone colour depends on the lighting conditions. Artificial light may create reddish pictures, as shown in Figure 3, which means different values for skin-tone colours. The histograms on the left side of figure 3 represent the distribution of skin hue and saturation components for artificial light (red line) and natural light (blue line). Values are shifted to the right for the artificial light values. A lighting compensation technique that uses "reference average" was introduced to normalize the colour appearance. The normalization operation subtracts from each pixel colour band (R,G,B) the average of the whole image, so odd coloured images like the reddish one are turned into more natural images. The histograms on the right side of figure 3 show that after this operation, skin-tone colours in different light conditions are much more similar.

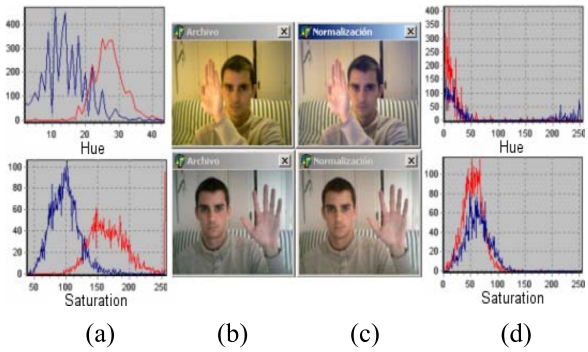


Figure 3. Skin detection: (a) Histogram for hue and saturation components before normalization operation, red line for artificial light and blue line for natural light; (b) original image under artificial and natural light conditions; (c) normalized image; (d) histogram for hue and saturation components after normalization operation.

Grouping Skin-Tone Pixels

Once the initial image has been smoothed and normalized, a binary image is obtained in which every white pixel represents a skin-tone pixel from the original image. The skin-tone classification is based on the normalized image and the considerations of the HSI space colour mentioned in section 3.1. Then, a pixel was classified as a skin-tone pixel if its hue and saturation components lay in a certain range. However, these ranges still vary slightly depending on light conditions, user's skin colour and background. These ranges are defined by two rectangles in the HS plane: the R_1 rectangle for natural light ($0 \le H \le 15$; $20 \le S \le 120$) and the R_2 rectangle for artificial light ($0 \le H \le 30$; $60 \le S \le 160$).

It is necessary to deal with wrongly classified pixels, not only with false positives but also with false negatives, once the binary-skin image has been computed. In order to remove background noisy pixels and fill small gaps inside interest areas, all 5x5 neighbourhoods are analyzed. The value of a certain pixel may change from skin to background and vice versa depending on the average amount of skin pixels in all 5x5 neighbourhoods.

The next step consists on the elimination of all those pixels that are not critical for shape comparison. It is not necessary the use of classical convolution operators because the image at this stage is a binary one, so edge borders were found leaving on the image just those pixels that had at least one background pixel on their neighbourhood. Optimal edge maps, where no redundant pixels could be found, were produced with the use of a 4-connectivity neighbourhood.

Blobs Analysis

Blobs, Binary Linked Objects, are groups of pixels that share the same label due to their connectivity in a binary image. After a blob analysis, all those pixels that belong to a same object share a unique label, so every blob can be identified with this label. Blob analysis creates a list of all the blobs in the image, along with global features: area, perimeter length, compactness and mass centre about each one. After this stage, the image contains blobs that represent skin areas of the original image. The user's hand may be located using the global features available for every blob, but the system must have been informed whether the user is right or left handed. Most likely, the two largest blobs must be the user's hand and face, so it will be assumed that the hand corresponds to the right most blob for a right-handed user and vice versa.

4. HAND POSTURE RECOGNITION

Once the user hand has been segmented, a model-based approach based on the Hausdorff distance that works on edge map images and a visual memory is proposed to recognize the hand posture.

Hausdorff Distance (HD)

The Hausdorff distance [Ruc96] is a measure between two sets of points. Unlike most shape comparison methods, the Hausdorff distance between two images can be calculated without the explicit pairing of points of their respective data sets, A and B. Formally, given two finite sets of points $A = \{a_1,..,a_m\}$ and $B = \{b_1,..,b_n\}$, the Hausdorff distance is defined as:

$$H(A,B) = \max\left(h(A,B), h(B,A)\right) \tag{1}$$

where

$$h(A,B) = \max_{a\in A}\min_{b\in B}\|a-b\| \tag{2}$$

The function $h(A,B)$ is called the *directed Hausdorff* distance from set A to B. It ranks each point of A based on its distance to the nearest point in B and then uses the largest ranked such point as the measure of distance. The Manhattan distance is used to define the distance between any two data points. In order to avoid erroneous results due to occlusions or noise conditions, the Hausdorff distance can be naturally extended to find the *best partial Hausdorff* distance between sets A and B. It is defined as:

$$h_k(A,B) = \underset{a\in A}{K^{th}}\min_{b\in B}\|a-b\| \tag{3}$$

Using the previous definition, the *Bidirectional Partial Hausdorff* distance is defined as follows:

$$H_{kl}(A,B) = \max\left(h_k(A,B), h_l(B,A)\right) \tag{4}$$

Matching Hand Postures using Hausdorff Distance (HD) and Visual Memory System (VMS)

A hand posture matching approach was developed by introducing the notion of a visual memory system and focusing on the Hausdorff distance introduced in section 4.1.

4.2.1 Visual Memory System

Because our problem approach is slightly different from the common one [San01, Bar98, Ruc96], different solutions are required. Our system has a visual memory (VMS) which stores recognizable pattern postures. In order to address diverse visual aspects for each stored pattern M and non-local distortions, the pattern postures are represented by q different samples:

$$VMS = \left\{P_{mq}^{in} : m = 1,2,\ldots,M; q = 1,2,\ldots,Q; i = 1,\ldots,3; n = 1,2,\ldots N\right\} \tag{5}$$

where each q^{th} sample of each m^{th} pattern hand posture is defined by its i^{th} binary edge map, its i^{th} distance transform [14] and its i^{th} morphologic information respectively, as follows:

$$P_{mq}^{1n} = \left\{p_{mq}^{11}, p_{mq}^{12}, \ldots, p_{mq}^{1N}\right\}; P_{mq}^{2n} = \left\{p_{mq}^{21}, p_{mq}^{22}, \ldots, p_{mq}^{2N}\right\}; P_{mq}^{3n} = \left\{p_{mq}^{31}, p_{mq}^{32}, \ldots, p_{mq}^{3N}\right\} \tag{6}$$

VMS can be created in a start-up stage, where the user introduces a set of specific hand gesture vocabulary. New postures can be added to the visual system whenever the user wants to. Furthermore, the hand gesture vocabulary could be obtained by a hand tracking system as the one proposed in [San99]. A detail of four hand posture patterns that composes VMS is shown in Figure 4.

Every J segmented user hand posture (UHP) is defined as:

$$UHP = \left\{U_j^{in} : j = 1,2,\ldots,J; i = 1,\ldots,3; n = 1,\ldots,N\right\} \tag{7}$$

In the same way of (6), each i^{th} value of U_j is defined by its binary edge map, distance transform and morphologic information.

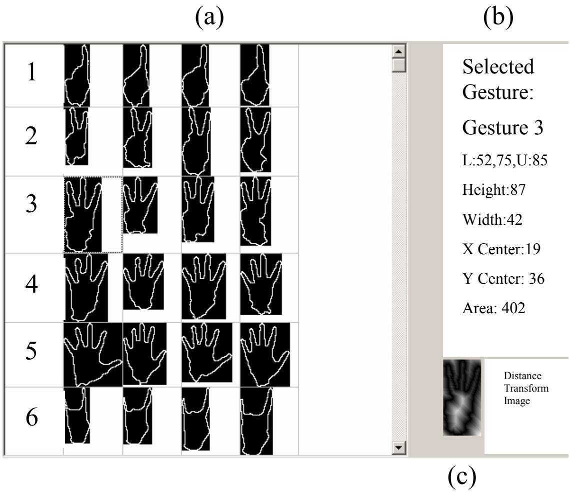


Figure 4. Detail of six stored hand postures of Visual Memory System (VMS). There are four samples for each posture (*q* = 4). Each one is represented by its edge map (a), morphologic information (b) and its distance transform (c).

4.2.2 *Matching Hand Postures*

The matching hand postures solution involves finding the minimum *bidirectional partial Hausdorff* distance (4) between U_j^{1n} and P_{Mq}^{1n}, where U_j^{1n} represents the j^{th} user hand posture edge map computed from the user hand posture detection module (described in section 3) and P_{Mq}^{1n} denotes each one of the *M* stored patterns edge map in VMS.

With the aim of considering the different changes in appearance and non-rigid distortions of user hand postures regarding the stored patterns, a resize operation is computed. Firstly, U_j^{1n} is scaled to the size of the q^{th} sample, which represents a stored pattern P_{mq}^{1n} in VMS and secondly, the specific pattern in VMS, P_{mq}^{1n} is scaled to the size of U_j^{1n}. Figure 5 illustrates this process in the steps 1 and 4. We denote both scaled operations respectively $S\left(U_j^{1n}\right)$ and $S\left(P_{mq}^{1n}\right)$.

In order to accelerate the computation of the *bidirectional partial Hausdorff* distance and make it suitable for real-time recognition, a pre-processing stage for each one of the stored pattern in VMS is calculated. The pre-processing step is based on the storage of the edge map, the distance transform and the morphologic information in the initialization process of the system (described in section 2).

The distance transform [Pag92] for each edge map posture image generates an image where background pixels have been labelled with their Manhattan distance to the closest object pixel.

The computation of the *bidirectional partial Hausdorff* distance consists of a seven-stage processing scheme, which is graphically showed in Figure 5. This process is repeated for every stored pattern that composes the VMS. Finally, the posture is classified as the pattern for which the minimum *bidirectional partial Hausdorff* distance is given.

Postures may not belong to the VMS, so a decision rule is used for rejecting them:

$$\boldsymbol{output} = \begin{cases} \boldsymbol{P_{m^*q}^{1n}} \ \boldsymbol{if} \ \boldsymbol{H_{kl}}\left(\boldsymbol{h_k}\left(\boldsymbol{S}\left(\boldsymbol{U_j^{1n}}\right), \boldsymbol{P_{m^*q}^{1n}}\right), \boldsymbol{h_l}\left(\boldsymbol{P_{m^*q}^{1n}}, \boldsymbol{S}\left(\boldsymbol{U_j^{1n}}\right)\right) \leq \alpha\right) \\ \boldsymbol{otherwise, reject} \end{cases} \quad (8)$$

Where $P_{m^*q}^{1n}$ is the classified pattern. This way, the output will be discarded if the minimum *bidirectional partial Hausdorff* distance if higher than a given threshold α.

The computation of the *partial directed distance* can be speeded up using the notion of the distance transform image. It is based on overlapping edge points of $S\left(U_j^{1n}\right)$ on P_{mq}^{1n}*'s* distance transform image. Then, for every edge point in $S\left(U_j^{1n}\right)$, the value in P_{mq}^{1n} is taken, and the *directed partial distance,* $h_k\left(S\left(U_j^{1n}\right), P_{mq}^{1n}\right)$, matches up with the k[th] quartile of those values. $h_l\left(S\left(P_{mq}^{1n}\right), U_j^{1n}\right)$ is computed in same fashion that $h_k\left(S\left(U_j^{1n}\right), P_{mq}^{1n}\right)$, replacing U_j^{1n} *by* P_{mq}^{1n} *and* P_{mq}^{1n} *by* U_j^{1n}. In each comparison stage, it is only necessary to compute the distance transform image of U_j^{1n}, since the one of P_{mq}^{1n} has previously been calculated and stored in VMS.

Figure 6(a) shows a detail of the comparison process, corresponding to the computation of $h_k\left(U_j^{1n}, P_{mq}^{1n}\right)$, where the user hand posture is compared with the distance transform image of one of the stored pattern in VMS. It should be noticed that, if the postures that are compared are similar, the size change does not affect gravely the shape of the posture, like observed in 6(a). If they are different, on the other hand, an image deformation can take place like observed in 6(b).

5. EXPERIMENTS

The hand posture detection and recognition approach was implemented in Borland Delphi on an AMD K7 700Mhz PC running Windows XP with an USB Logitech Quickcam webcam. The recognition approach has been tested with real world images as the input mechanism of a user interface, surveillance system, autonomous robot vision system or virtual environment applications. All tests were executed on 128x96 images with 24b colour depth, which were sampled at a rate of 25 frames/sec.

The system was tested under two different light conditions: natural daylight and artificial light, in order to test its adaptability to different lighting conditions. Also, two different users were considered. The first one, named “original subject” was the one who created the visual memory. The second one tested the application using the same visual memory created by the original subject. The gesture vocabulary in our gesture interface is composed by 26 postures, shown in Figure 7, each of which represent a gesture command mode. It can be observed that some postures are really similar, like postures pair 1-20, 2-21, 5-25, 18-19, 21-22.

Recognition Performance and Discussion

In order to test system recognition performance, 100 frames were processed for each posture in the visual memory (VMS), and hree outputs were considered: “*right*” meant a correct classification; “*discarded*” meant a posture which did not belong to the visual memory (VMS), and “*wrong*” was an incorrect classification. The graphic illustrated in Figure 8 shows the average classification output under artificial and natural light conditions for the original subject and the results under artificial light conditions for the second subject. The system reaches a 95% recognition rate under optimal circumstances (same visual memory user and artificial light conditions). Even under different light conditions (natural light) and being the results of the segmentation process slightly different, the system is able to reach an 86% recognition rate (figure 8b).

In order to study the system’s adaptability to hand morphologic changes, the test was repeated with a different user. Results are shown in figure 9. Although reasonably high, these results imply that each user should create their own visual memory, in same fashion as voice recognition applications must be trained for each user. In these situations, it has been observed that better results are obtained if the samples for each stored posture in the visual memory (VMS) are generated using more than one user, because a certain degree of variability is added to the sample.

Results show a high recognition rate, where the system can achieve a 90% recognition average rate. It can be affirmed that if a posture edge map is properly segmented in the image processing stage, the Hausdorff matching approach will classify it properly. What is more, posture pairs which are really similar, like 1-20, 2-21, 5-25 and 21-22 in figure 7, are properly classified.

Figure 10 shows the required time to process each of the stages of the developed approach. An amount of 500 frames were analysed in order to obtain an average of the time needed for each stage processing.

The average processing time per second on an AMD K7 700Mhz PC system is 8 frames. Being impossible for a human to make 25 postures in a second time, it is feasible the analysis on just 8 frames. It is noticed that this feature is needed to be improved; a tracking system should be implemented in order to avoid the complete analysis of each frame. Anyway, the use of a faster PC system than the currently used would assure a real time processing, which allows the use of the proposed approach in real-time video applications.

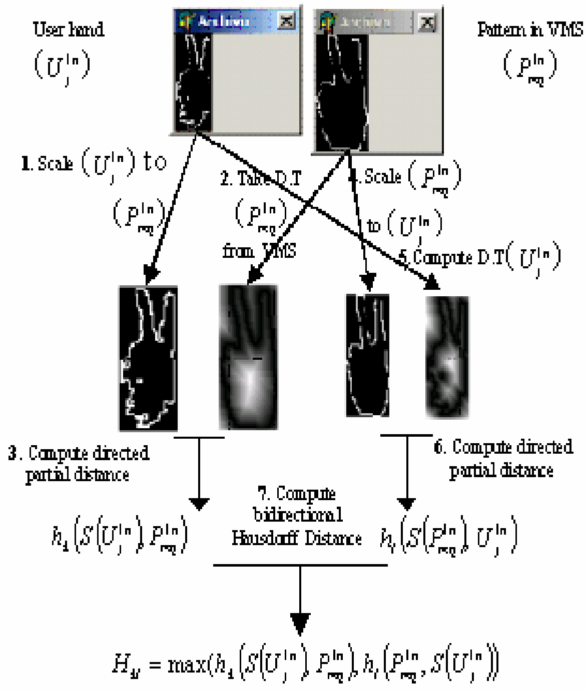


Figure 5. Computation of *bidirectional partial Hausdorff* distance for hand posture recognition problem. D.T denotes the distance transform image.

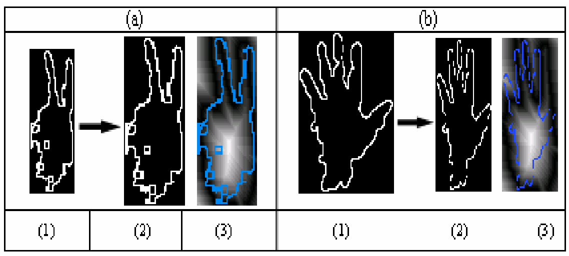


Figure 6. Shape Comparison between user hand posture and a stored pattern in VMS: (a) similar postures, (b): deformed postures, (1): user hand posture edge map, (2): specific pattern edge map in VMS, (3): overlapping user hand posture edge map (in blue) on pattern’s distance transform image.

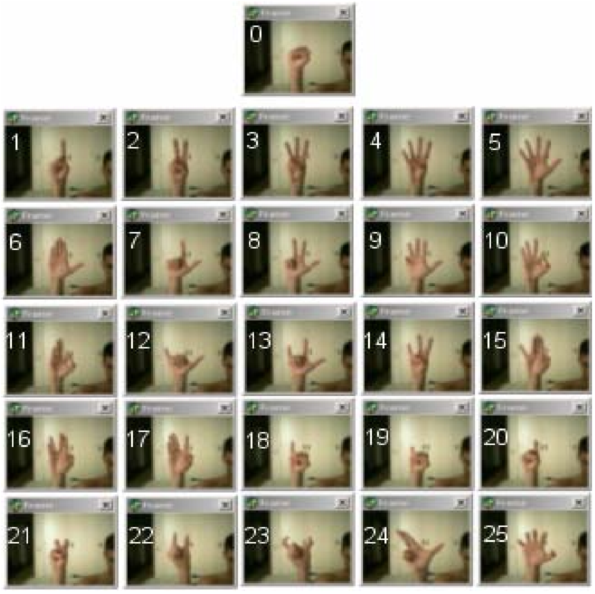
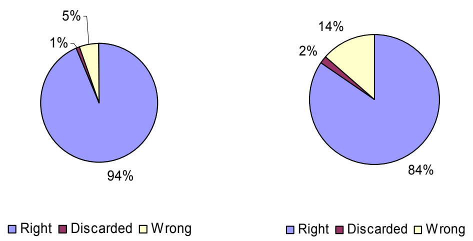


Figure 7. Posture Data numerated consecutively between 0 and 25.



(a) **(b)**

Figure 8. Recognition Rates: (a) original subject, artificial light; (b) original subject, natural light.

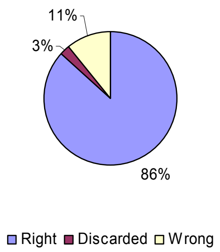


Figure 9. Recognition Rates: second subject, artificial light.

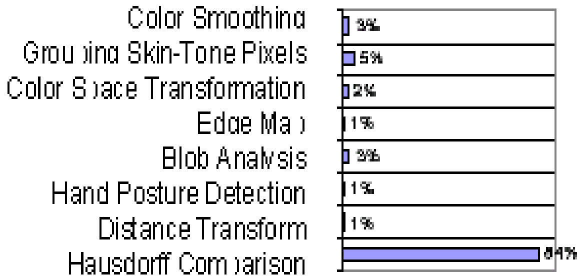


Figure 10. Time distribution measured in percentage to process each one of the stages of the detection and recognition hand posture approach.

6. CONCLUSIONS AND FUTURE WORK

A fast processing process and a robust matching carried out through a Hausdorff distance approach; a visual memory system and resolution of non-rigid distortions have been presented for hand posture detection and recognition problem. Different light conditions, backgrounds and human users have been tested in order to evaluate system's performance. The recognition rates show that the system is robust against similar postures. Even more, the runtime behaviour allows the use in real-time video applications with a simple personal computer and a standard USB camera.

Future research will concentrate on investigating efficient hierarchical *N*-template matching and studying other robust and efficient methods about face and hand location in order to integrate the components of the system into a gesture interface for an anthropomorphic autonomous robot with an active vision system [Her99] and into virtual environment applications.

7. ACKNOWLEDGMENTS

This work has partially been supported by the Canary Autonomous Government under Project PI2000/042.

8. REFERENCES

[Bar98] Barnabas Takacs. Comparing Face Images using the modified Hausdorff distance. Pattern Recognition, Vol 31, nº 12, pp. 1873-1881, 1998.

[Bob95] Bobick, A. And Wilson, A. A state-based technique for the summarization and recognition of gesture. In Proc. IEEE Fifth Int. Conf. on Computer Vision, Cambridge, pp. 382-388, 1995.

[Dam97]A. Van Dam. Post-WIMP user interfaces. Comunications of the ACM, VOL. 40, nº2, pp. 63-67, February 1997.

[Dav94] Davis, J. and Shah, M. Visual gesture recognition. IEE Proc. Vis. Image Signal Process, 141(2) :101-106, 1994.

[Her99]M. Hernández, J. Cabrera, M. Castrillón, A. Domínguez, C. Guerra, D. Hernández, J. Isern. DESEO: An Active Vision System for Detection, Tracking and Recognition. Volume 1542 of Lecture Notes in Computer Science, 1999.

[Hun95]. E. Hunter, J. Schlenzig, and R. Jain. Posture Estimation in Reduced-Model Gesture Input Systems. Proc. Int´l Workshop Automatic Face and Gesture Recognition, pp. 296-301, 1995.

[Jai95] R. Jain, R. Kasturi, B. Schunck. Machine Vision. McGraw-Hill Editions, 1995.

[Jon98] M. Jones and J. M. Rehg. Statistical Colour Models with Application to Skin Detection. Technical Report Series, Cambridge Research Laboratory, Dec, 1998.

[Ju97] Ju, S., Black, M., Minneman, S., Kimber, D.: Analysis of Gesture and Action in Technical Talks for Video Indexing. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR97.

[Mag95] C. Maggioni. Gesturecomputer. New Ways of Operating a Computer. *Proc.* Int´l Workshop Automatic Face and Gesture Recognition, 1995.

[Ruc96] Ruckelidge, W. Efficient Visual Recognition Using the Hausdorff Distance, volume 1173 of Lecture notes in computer science, Springer-Verlag, 1996.

[Pag92] Paglieroni, D.W. Distances Transforms. Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing, 54 :56-74, 1992

[Pen98]Pentland, A. Smart Rooms: Machine Understanding of Human Behavior. Computer Vision for Human-Machine Interaction, eds. Cambridge University Press, pp. 3-21, 1998.

[San01] Sánchez-Nielsen Elena, Lorenzo-Navarro Javier, Hernández-Tejera F. Mario. Increasing Efficiency of Hausdorff Approach for Tracking Real Scenes with Complex Enviroments. 11th IEEE International Conference on Image analysis and processing, pp. 131-136, 2001.

[San99] Sánchez-Nielsen Elena, Hernández-Tejera F. Mario. Hand Tracking using the Hausdorff Distance. Proceedings of the IV Ibero American Symposium on Pattern Recognition, 1999.

[Tri01] J. Triesch and C. von der Malsburg. A System for Person-Independent Hand Posture Recognition against Complex Backgrounds. IEEE, 2001 Transactions on Pattern Analysis and Machine Intelligence, Vol.23, Nº. 12, December 2001.

[Turk98] M. Turk, (ed.). Proceedings of the Workshop on Perceptual User Interfaces, San Francisco, CA, November 1998.

[Wil95]William T. Freeman, Craig D. Weissman. Television Control by Hand gestures. IEEE Intl. Workshop on Automatic Face and Gesture Recognition, Zurich, June, 1995.