U.P.B. Sci. Bull., Series D, Vol. 66, No. 2-4, 2004

STATIC HAND GESTURE RECOGNITION SYSTEM

B. IONESCU, D. COQUIN, P. LAMBERT, V. BUZULOIU*

Acest articol discută despre folosirea tehnicilor "computer vision" la interpretarea gesturilor umane. Este propus un sistem de recunoaştere a gesturilor statice. Drept aplicaţii ale acestui sistem, putem menţiona ghidarea robţilor sau interacţionarea cu anumite dispozitive "hardware". Imagini cu nivele de gri ale poziţiei mâinii sunt capturate cu un aparat digital. Recunoaşterea gesturilor este realizată in trei etape. In etapa de pre-procesare a imaginilor este izolată regiunea corespunzătoare mâinii. In etapa de parametrizare, pe baza skeletonului regiunii mâinii, se calculează măsuri ale orientarii si poziţiei degetelor si a palmei. In etapa de clasificare, folosind alfabetul de gesture, se determină gestul realizat. Sunt prezentate şi rezultate experimentale.

This paper deals with the use of computer vision for interpreting human gestures. A static gesture recognition system is proposed. As applications of this system we can mention: guiding robots or interacting with hardware devices. Images of hand postures are acquired using a single grey level digital camera. The gesture recognition is performed in three steps. In the preprocessing step the hand region is extracted. The parameterization step extracts the hand region skeleton and computes measurements of fingers-palm-thumb positions and orientation. The classification step uses the gesture alphabet in order to determine which gesture was performed. We also present some experimental results.

Keywords: hand gesture recognition, binary image, skeleton, Chamfer distance.

Introduction

Within the last years, in the field of "Computer Vision", considerable resources have been allocated in the developing face and hand recognition techniques. Being able to recognize hand and face region from static images or video sequences gives us an advantage in application such as teleconferences, telemedicine and in the field of human-computer interaction devices. In particular, the hand gestures are an attractive way of interacting with such systems because these are the natural ways of communication. Another advantage is that the user does not have to learn how to manipulate certain specialized hardware devices.

* PhD student, The Image Processing and Analysis Laboratory, University "POLITEHNICA" of Bucharest, Romania; Associate Professor, Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance, Université de Savoie, France; Prof., The Image Processing and Analysis Laboratory, University "POLITEHNICA" of Bucharest, Romania.

In the field of hand gesture analysis we can distinguish three main approaches: **glove-based analysis**, **vision-based analysis** and **analysis of drawing gestures** [1]. Glove-based analysis requires special devices such as glove-based devices which employ some kind of sensor (mechanical or optical) attached to a glove that transduces finger flexion and abduction into electrical signals. Vision-based analysis is the most natural way of constructing a human-computer gesture interface. It is based on the major way humans perceive information about their surroundings: images. Yet it is also the most difficult one to implement in a satisfactory manner because of the limitations in machine vision today. The vision-based analysis is performed by using one or more cameras. The visual information about a person in some visual environment is acquired and the necessary gesture is extracted. However there are some difficult problems: the segmentation of the hand or the recognition of the hand posture. To facilitate these operations one can use markers, marked gloves or restrictive setups: uniform background, very limited gesture vocabulary or just a simple posture analysis. Drawing gestures can be regarded as those gestures aimed at commanding a computer through a sequence of hand strokes. It usually involves the use of a hand held device like a stylus or a computer mouse as an input device. The gestures are interpreted using the data provided by the input device.

The process of hand gesture recognition can be divided into two main tasks, as following: the **feature extraction** where low-level information from the raw data is analyzed to produce higher-level semantic information and the **classification** where the data collected will be used to detect the hand gesture from a gesture alphabet [1].

The feature extraction techniques can be divided in: **simple feature extraction** techniques where the features are based on distance, velocity and acceleration information, energy measurements or angle information; **active shape models** where the hand edges are detected and tracked; **principal component analysis** which is a statistical technique of reducing the initial data set to a smallest and non-redundant one and **spatio-temporal vector analysis** which uses the shape of the hand to extract invariant moments [2][3][4].

As classification techniques we can mention: the **template-based approach** with conventional template matching, instance-based learning [5], the linguistic approach [6] and appearance-based motion analysis [7]; **statistical methods** (hidden Markov models [8]) and **miscellaneous algorithms** (neural networks and causal analysis [9][10]).

The most used classification method is the conventional template matching, which is generally the simplest way and it is performed in two steps. In the first part, the templates are created by collecting data values for each hand posture in the posture set. Generally each hand posture is performed a number of times and the average of the raw data for each sensor (camera or glove) is taken

and stored as template. The second part consists in comparing the current sensor readings with the given set of templates to find the posture template most closely matching the current data record.

Hand gestures could be classified as **static hand gestures** which are characterized by a hand posture with a particular finger-thumb-palm configuration, and **dynamic hand gestures** characterized by the initial and final stroke hand configuration and the general stroke motion [11]. Static gestures are represented by a single image instead of an image sequence as in the case of dynamic gestures.

The hand gesture recognition system proposed in this article aims at recognizing static hand gestures. As feature extraction technique, the simple feature extraction technique will be used, in order to extract finger and palm orientation and relative positions. The gesture alphabet is apriori known. The classification will be performed using the conventional template matching approach due to its effectiveness.

1. The description of the recognition system

The proposed static hand gesture recognition system is developed for recognizing a small gesture alphabet which represents the commands for manipulating a hi-fi device such as a stereo set, as already mentioned. The gesture alphabet is apriori known. The system diagram is presented in Fig. 1.

Grey level images of the hand posture are acquired with a digital camera. In order to facilitate the hand region extraction we assume that the gestures are performed against a uniform background, darker than the hand region. The images are preprocessed in order to extract the hand region. From the initial hand posture grey level image a binary image is computed using histogram segmentation techniques enhanced afterwards using mathematical morphology and hole filling techniques. The result will be a binary image with the hand region isolated from the background.

The feature extraction step consists in the hand region skeleton computation from the obtained binary image and its parameterization. The skeleton is a compact representation of the hand region preserving its topology. The features extracted from the skeleton are the finger and thumb relative orientations and size. Different gestures have different finger and palm orientations.

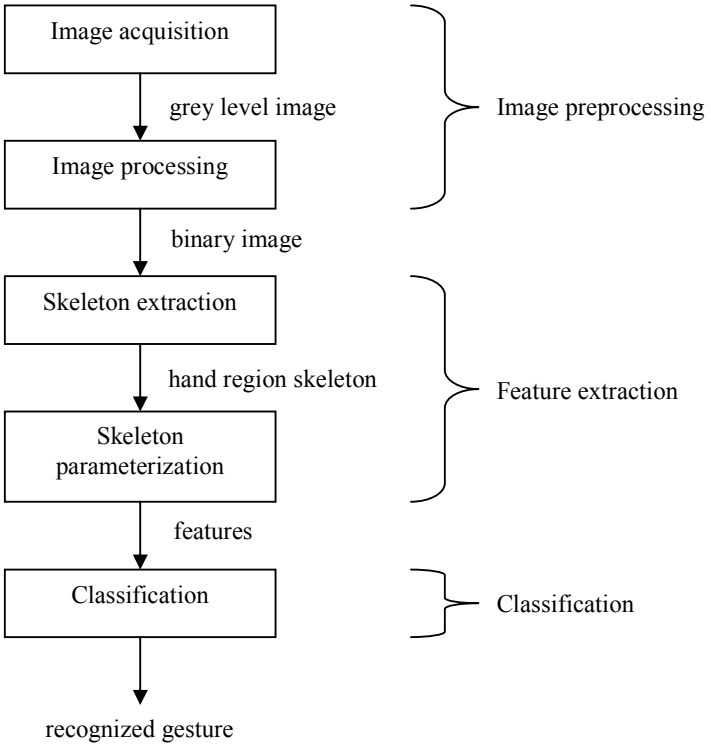

Fig.1. Static hand gesture recognition system diagram.

The classification step uses the data obtained from the feature extraction step and compares them with the apriori know data for the gesture alphabet. The recognized gesture is the gesture with the similar features.

2. Image preprocessing

As already mentioned, the image preprocessing step consists in isolating the hand region from the background. The preprocessing chain will have the following steps: the binary image computation, the binary image enhancement and the hand region extraction.

A binary image is an image with only two levels of grey, usually 0 and 255. The computation is done by reducing the grey levels from the original image using the histogram segmentation [12]. The grey level histogram is defined as

$$h(i) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \delta(i - f(m,n)) \tag{1}$$

where f() is the grey level image, MxN is the image size in pixels, L is the number of grey-levels (usually 256), δ(x-y) is 1 if x=y and 0 otherwise, i=0..L-1 and h() is a vector of size L.

Taking into account our restrictive background, our image will contain only two objects: the hand region and the background. This fact will lead to a bimodal histogram, one mode corresponding to the hand region and the other to the background (see Fig. 2). The binary image computation consists in the thresholding [12] the image with the threshold T placed between the two modes of the histogram. To reduce the steep variations of the histogram, a 1D mean filter will be performed. The binary image g() will be computed as following

$$g(m,n) = \begin{cases} E_0 & 0 \le f(m,n) < T \\ E_1 & T \le f(m,n) < L \end{cases} \quad (2)$$

where g() is the binary image, f() is the grey level image, L is the number of grey levels, T is the threshold, E_0 and E_1 are the regions labels (0 for background and 255 for objects), MxN is the image size in pixels and m=0..M, n=0..N (an example is shown in Fig. 3.b).

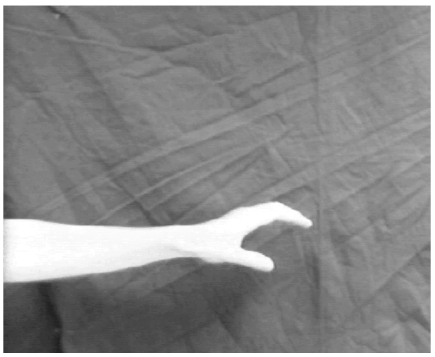

Fig.2.a. Hand posture grey level image

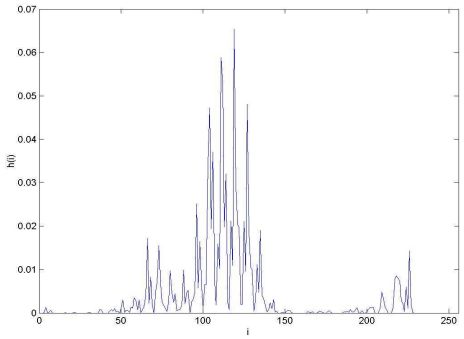

Fig.2.b. Histogram (oX grey level index, oY pixel count)

Because of the fact that the two modes are overlaid, some pixels corresponding to background will be determined as hand region pixels and vice versa. This problem will be solved using mathematical morphology and convex regions analysis [13]. The digital mathematical morphology deals with the study of the topologies and the structure of the objects. The hand region obtained can be considered as well as an object. As morphological operations, an opening filter followed by a closing filter [13] will be used to smooth the contours and close the objects (see Fig. 3.c). Another step for the binary image enhancement is to fill the small holes that may appear in the hand region. A hole can be defined as a small region of background pixels surrounded by object pixels. The proposed algorithm

inspects for each background pixels its eight neighbors situated in the eight cardinal points at a distance of d pixels from the current pixel. Decreasing the distance d, if all neighbors are object pixels, then all the background pixels contained in the discrete disc of ray d and centered in the current analyzed pixel become object pixels (see the result in Fig. 3.d).

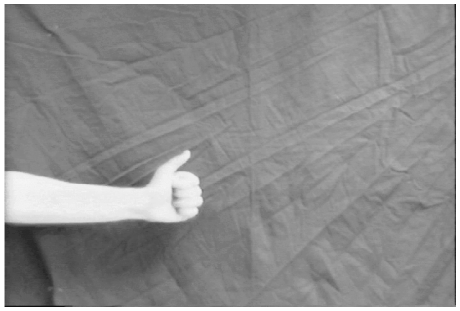

Fig.3.a. Hand gesture grey level image

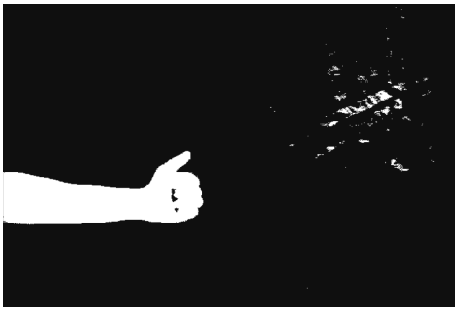

Fig.3.b. Binary image after histogram segmentation

Fig.3.c. Binary image after morphological enhancement

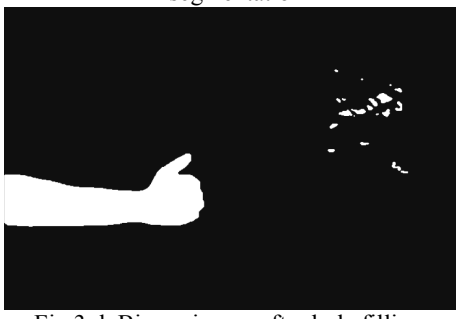

Fig.3.d. Binary image after hole filling

Fig.3.e. Convex regions labeling

Fig.3.f. The resulting hand region

Hand region extraction is then performed by the analysis of all connected regions of pixels. Small regions of pixels may appear as a result of a non-uniform illumination of the scene (see Fig. 3.b). The hand region is the biggest connected region of object pixels. We define the area of a region as the number of object

pixels. Each area will be determined using a flood fill algorithm (see Fig. 3.e, region labeling). An example of hand region extraction can be seen in Fig. 3.

3. Feature extraction

In the feature extraction step, a set of parameters of the hand region skeleton will be computed. As already mentioned, a skeleton is a compact representation of an object (the hand region in the binary image, in our case). The desired properties of the skeleton are: preserving the topology of the object (the same number of connected components and the same number of holes), robust against translations, rotations and scaling, thin (at most 2 pixels wide). Many methods and algorithms have been proposed to satisfy most of the skeleton properties: analytical methods, morphological methods, thinning methods and distance transformation based methods [14].

Fig.4.a. Hand region binary image

Fig.4.b. Distance image (the value of each object pixel represent its distance to the closest background pixel)

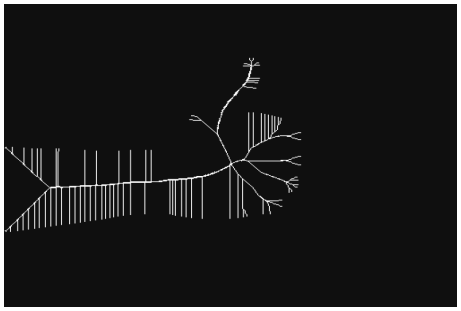

Fig.4.c. Connected local maxims

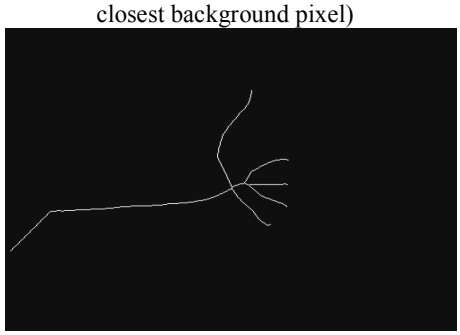

Fig.4.d. The resulting skeleton after the pruning and beautifying step

The extraction of the hand region skeleton will be performed using the Chamfer distance transformation method as described in [14]. An example of a skeleton extraction is presented in Fig. 4. First of all the binary image of the hand region is transformed in a distances image using the Chamfer distance

transformation (see Fig. 4.b). On the distances image the median axis is extracted by extracting and connecting the local maxima (see Fig. 4.c). The resulted skeleton may have undesired holes and branches. The holes filling and the pruning step yield a skeleton which is more robust with respect to geometrical operations (i.e. rotations, translations). A "beautifying" step is also used to eliminate some spurious pixels and to improve the skeleton shape [14] (see Fig. 4.d).

The obtained skeleton will be used for the hand gesture feature extraction. For each gesture within the current gesture alphabet a set of parameters based on the geometrical properties of the skeleton will be computed. As a spatial reference, the gravity mass center G of the hand region will be used as the center of the Cartesian XoY system. Based on that, the image will be divided into four regions, each one corresponding to a quadrant (see Fig. 5).

If $G(m_c,n_c)$ is the mass center, its coordinates are given by

$$m_c = \frac{1}{N_p}\sum_{m=1}^{M}\sum_{n=1}^{N}\delta(g(m,n)-255)\cdot m \tag{3}$$

$$n_c = \frac{1}{N_p}\sum_{m=1}^{M}\sum_{n=1}^{N}\delta(g(m,n)-255)\cdot n \tag{4}$$

where N_p is the total number of object pixels, g(m,n) is the binary image of the hand region of size MxN pixels (an object pixel has the value 255) and δ(x-y) is 1 if x=y and 0 otherwise.

Using as reference the gravity mass center G, the position of some important points will be analyzed (see Fig. 5). We define the following particular points:

- **terminal point**: a marginal branch point (the end of a skeleton branch)
- **intersection**: a skeleton point where three or more branches converge
- **segment**: the sequence of skeleton pixels between a terminal point and an intersection point.

The feature set depends on the hand gesture alphabet. As a feature set the position of the terminal points, the size and the orientation of the segments will be used. The segments orientation will be approximated with the orientation of the straight line which connects its terminal and the intersection points (see Fig. 5).

Based on the data collected (the feature set), a set of rules which characterize each gesture is extracted. If ϖ is the collection of all rules, $\varpi = \{\varpi_1,...,\varpi_n\}$, where ϖ_i is the set of rules for the gesture I, then $\varpi_i \cap \varpi_j = 0$ for $i \neq j$ and $\bigcup_{i=1}^{n}\varpi_i = \varpi$.

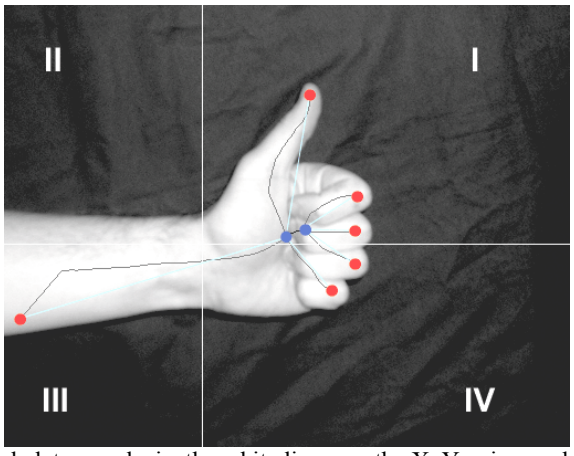

Fig.5. Example of skeleton analysis: the white lines are the XoY axis, quadrants are numbered from I to IV, the intersections and terminal points are marked with circles and t he segments are approximated with straight lines.

For example, the gesture illustrated in Fig. 5 has the longest segment with an orientation of approximately 180 degrees and it is contained in the third quadrant, and the second longest segment has an orientation between 45 and 90 degrees being contained in the first quadrant.

4. Gesture classification

Once the features for all the gestures within the gesture alphabet are determined and stored, the system is ready for the gesture recognition task. A new gesture is analyzed as described in the system diagram (see Fig. 1). The result will be its set of features (terminal points and intersections, segment positions and size etc). The new obtained data are compared with the data stored and the most similar gesture is pointed as the performed gesture.

5. Experimental results

The proposed static gesture recognition system was tested on a small gesture alphabet containing five distinct gestures, which are the commands of a classic stereo set: “stop”, “play forward”, “play backward”, “increase volume” and “decrease volume” (see Fig. 6).

The hand posture images were taken with a grey level digital camera. The hand gestures are performed in a certain space area and against a darker background, as we can see in Fig. 6.

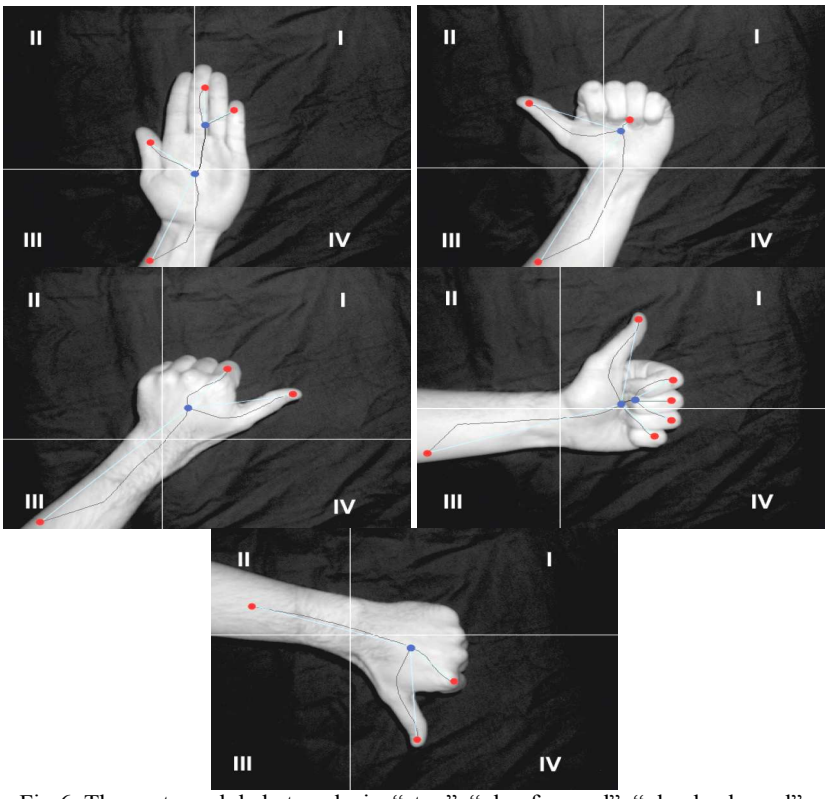

Fig.6. The gesture alphabet analysis: “stop”, “play forward”, “play backward”, “increase volume” and “decrease volume” (from left to right and up to down)

For our gesture alphabet, the following set of rules was determined based on the position of the terminal points and the size of the segments on the hand region skeleton (see Fig. 6 and table 1).

Table 1.

Gesture classification rules for the gesture alphabet

Gesture	Position of the longest segment (quadrant)	Position of the second longest segment (quadrant)	Approximate angle of the second longest segment (degrees)
“play forward”	III	II	150-180
“play backward”	III	I	0-45
“increase volume”	III	I	50-90
“decrease volume”	II	IV	280-320
“stop”	all the skeleton intersections are near the oX axis		

The set of rules was determined based on the observation that each gesture is characterized by a particular finger orientation. The finger corresponds to the

second largest segment of the hand skeleton. Each gesture also assumes a different position of the finger within the four quadrants. The “stop” gesture always has all the intersections points very close to the oX axis because of the symmetry of the hand (see Fig. 6).

The system was tasted offline with four different sets of images of all the gestures within the gesture alphabet (20 hand posture images). The gesture recognition starts with the analysis of the longest and the second longest segment in the hand skeleton. If the new gesture follows one of the mentioned rules (see table 1) the recognition is successful; if not, the system checks the rules for the “stop” gesture. A 100% recognition rate was achieved. The perfect recognition ration is a result of the fact that the gestures are dissimilar and the hand region skeletons cannot be mistaken. The only recognition errors can appear because of the skeleton computation (holes or undesired branches in the skeleton).

Conclusions

In this paper we propose a static hand gesture recognition system. Images of the hand gesture are captured with a grey level digital camera afterwards processed by a computer. In order to facilitate the hand region extraction, a restrictive scene is used; the gesture is performed against a darker background and in a certain position in the space.

The hand region is extracted from the grey level image and its skeleton is computed using a distance based transformation. The geometrical properties of the skeleton (terminal points, intersections, segments and their positions) are then analyzed in order to define a set of rules for recognizing the performed gesture.

The gesture alphabet to be recognized is apriori known and the set of rules are manually determined and introduced in the processing software. Changing the gesture alphabet means changing the recognition rules.

When the recognition process begins, the parameters of the new gesture are compared with the ones stored in the gesture alphabet and the gesture classified.

The system was tested offline on a small gesture alphabet containing commands for a stereo set (“stop”, “play forward”, “play backward”, “increase volume” and “decrease volume”) and it achieved a perfect recognition ratio.

The advantage of such a system consists in the accuracy of making the distinction between hand gestures that have different finger configurations, and it is suited for a small gesture alphabet. Gestures that imply similar finger-palm-thumb configuration will lead to similar skeletons which are difficult or impossible to distinguish.

The system is unable to recognize gestures that do not have a distinct finger orientation (like “stop” gesture), which are hard to be recognized because the skeleton does not present distinctive branches.

For practical systems using the proposed algorithm one should use dedicated hardware/DSP implementation for achieving real time processing. This is one of the aims of the future development we envisage.

REFERENCES

1. *V. Pavlovic, R. Sharma, T. S. Huang*: Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review, in IEEE Transactions on Pattern Analysis and Machine Intelligence, **19**(7), 1997, pp 677-695
2. *Ho-Sub Yoon, Jung Soh, Younglae J. Bae, Hyun Seung Yang*: Hand gesture recognition using combined features of location, angle and velocity, in Pattern Recognition, **34**, 2001, pp 1491-1501
3. *S. Nayaga, S. Seki, R. Oka*: A Theoretical Consideration of Pattern Space Trajectory for Gesture Spottin Recognition, in proceedings of International Conference on Automatic Face and Gesture Recognition, 1996, pp 72-77
4. *B. Raytchev, O. Hasegawa, N. Otsu*: User-independent Online Gesture Recognition by Relative Motion Extraction, in Pattern Recognition Letters, **21**, 2000, pp 69-82
5. *F. Quek.* Unencumbered Gestural Interaction, in IEEE Multimedia, **4**(3), 1994, pp 36-47
6. *H. Chris, I. Sexton, M. Mullan*: A Linguistic Approach to the Recognition of Hand Gestures, in proceedings of the Designing Future Interaction Conference, 1994
7. *Davis, J. William, M. Shah*: Gesture Recognition, Technical Report, Department of Computer Science, University of Central Florida, CS-TR-93-11, 1993
8. *F. S. Chen, C. M. Fu, C. L. Huang*: Hand Gesture Recognition Using a Real Time Tracking Method and Hidden Markov Models, in Image and Vision Computing, **21**, 2003, pp 745-758
9. *R. Kjeldsen, J. Kender*: Towards the Use of Gesture in Traditional User Interface, in proceedings of International Conference on Automatic Face and Gesture Recognition, 1996, pp 66-71
10. *C. W. Ng, S. Ranganath*: Real-time Gesture Recognition System and Application, in Image and Vision Computing, **20**, 2002, pp 993-1007
11. *Thomas S. Huang* and *Vladimir I. Pavlovic*: Hand Gesture Modeling, Analysis and Synthesis, in proceedings of International Workshop on Automatic Face and Gesture Recognition, **31**, 1995, pp 73-79
12. *Linda G. Shapiro, George C. Stockman*: Computer Vision, Prentice-Hall, Upper Saddle River, New Jersey, 2001, pp 39-48
13. *A. K. Jain*: Fundamentals of Digital Image Processing, Prentice-Hall, Englewood Cliffs, N.J.07632, 1989, pp 384-390
14. *Y. Chehadeh, D. Coquin, Ph. Bolon*: A Skeletonization Algorithm using Chamfer Distance Transformation Adapted to Rectangular Grids, in proceedings of IEEE International Conference on Pattern Recognition, **2**, 1996, pp 131-139.