Improvement in Recognition Techniques for Human Computer Interaction

Prof. Raju M. Tugnayat
Head of Department Information Tech.
Jawaharlal Darda Inst. of Engg. & Tech.
Yavatmal. (M.S.) India

Prof. Dr. Nitin K. Choudhary,
Principal
Shri Bhagavati College of Engg.
Nagpur (M. S.) India

M.V.Sarode
Asstt. Professor
Jawaharlal Darda Inst. of Engg.
& Tech Yavatmal. (M.S.) India

ABSTRACT

Hand gestures are an important modality for human computer interaction (HCI) [1]. Compared to many existing interfaces, hand gestures have the advantages of being easy to use, natural, and intuitive. Successful applications of hand gesture recognition include computer games control [2], human-robot interaction [3], and sign language recognition [4], to name a few. Vision-based recognition systems can give computers the capability of understanding and responding to hand gestures. The usability of such systems greatly depends on their ability to function reliably in common real-world environments, without requiring the user to wear special clothes or cumbersome devices such as colored markers or gloves [4]. The aim of this technique 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 use of a visual memory (Stored database) 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. A hierarchical gesture recognition algorithm is introduced to recognize a large number of gestures. Three stages of the proposed algorithm are based on a new hand tracking technique to recognize the actual beginning of a gesture using a Kalman filtering process, hidden Markov models and graph matching. Processing time is important in working with large databases. Therefore, special cares are taken to deal with the large number of gestures.

Keywords

Postures, HSI space, RGB Space, Eigen Space, Kalman Filtering.

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 [6] [7]. Moreover In order to improve the lip-reading efficiency Dr. Cornett developed the Cued Speech [8]. He proposed to add manual gestures to lip shapes so that each sound has an original visual aspect. Such a "hand & lip-reading" becomes as meaningful as the oral message. This technique aim to automatically recognize in real-time a succession of Cued Speech gestures. By coupling such a device with an automatic lip-reading module and others various automates, a complete hearing-impaired translator could be feasible. 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. A new vision-based framework is presented in this method, 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 (stored database). The recognition problem is approached through a matching process in which the segmented hand is compared with large number of the postures in the system’s memory using the new hierarchical algorithm, which is based on a dynamic model of hand movements, hidden Markov models and Graph matching. 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 or they can be generated during the recognition process.

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. The following processes compose the general framework:

a) **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.

b) **Acquisition**: a frame from the webcam is captured.

c) **Segmentation**: each frame is processed separately before its analysis: the image is smoothed, skin pixels are labeled, 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.

d) **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 three stage recognition process Kalman filtering process, hidden Markov models and graph matching.

e) **Executing Action**: finally, the system carries out the corresponding action according to the recognized hand posture the output of the system can then letter / digit / voice / word.

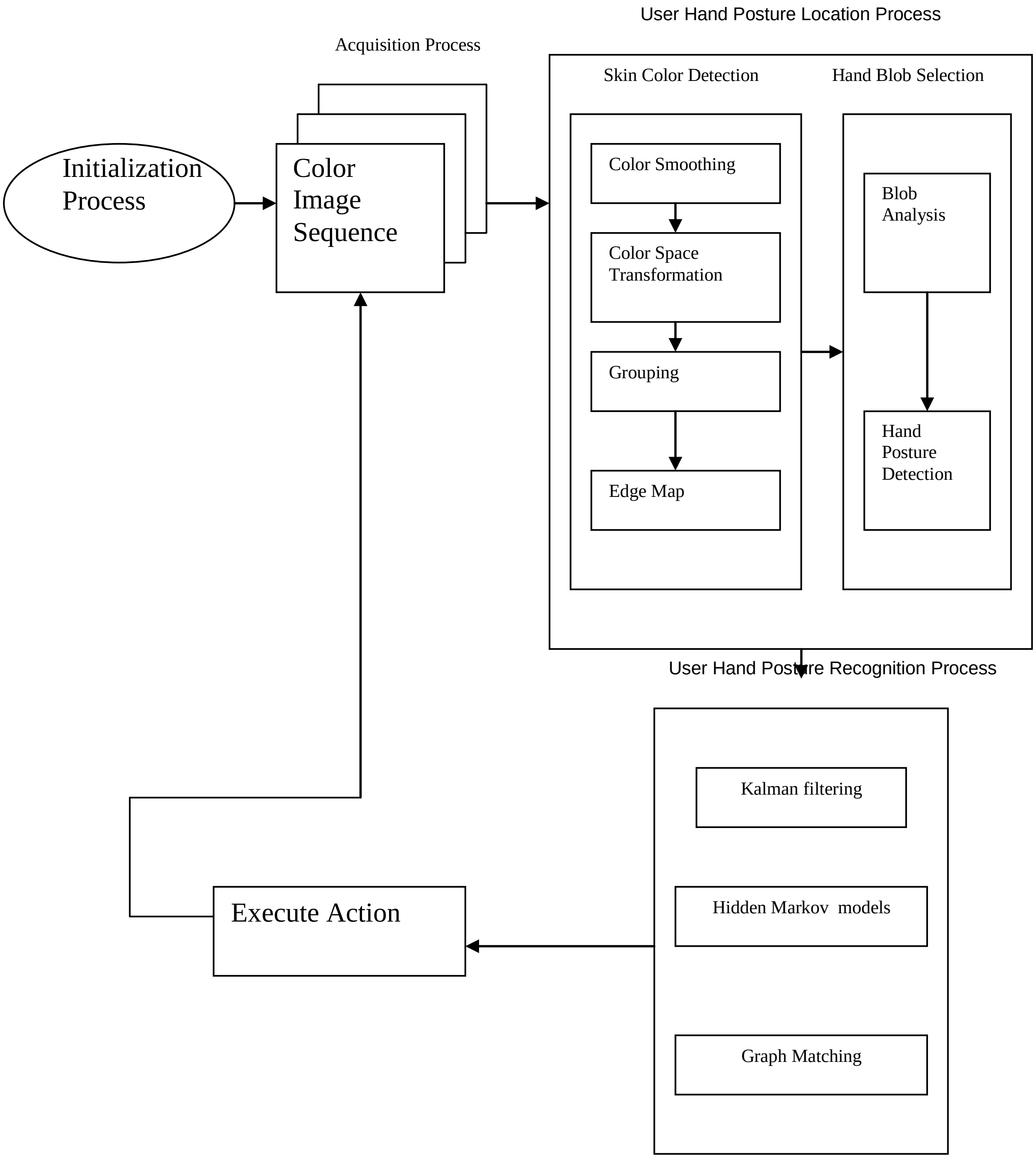

Figure 1. Global Hand Posture Detection and Recognition

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 Color Features

Modeling skin color requires the selection of an appropriate color space and identifying the cluster associated with skin color in this space. HSI space (Hue, Saturation and Intensity) was chosen since the hue and saturation pair of skin-tone colors are independent of the intensity component Thus, colors can be specified using just two parameters instead of the three specified by RGB space color (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 color features. Yellow dots represent samples of skin-tone color from segmented images, while blue dots are the rest of the image color 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 color 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.

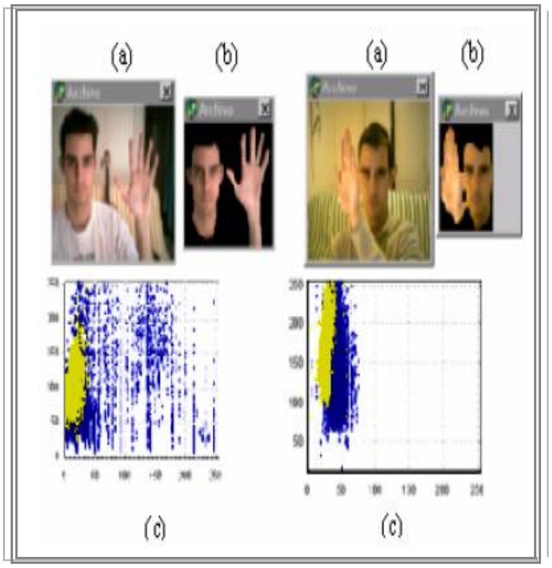

Figure 2. Skin-tone colour distribution in HIS space: (a) original image under natural and artificial light conditions; (b) segmented- skin image; (c) HSI colour space (yellow dots represent skin colour samples and blue dots represent the rest of the image samples), x-axis for hue component and y-axis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

Color 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 colors. Best results were achieved using a mean filter, among the different approaches of proposed lineal filters. The appearance of the skin-tone color depends on the lighting conditions. Artificial light may create reddish pictures, as shown in Figure 3, which means different values for skin-tone colors. 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 color appearance. The normalization operation subtracts from each pixel color band (R,G,B) the average of the whole image, so odd colored 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, skintone colors in different light conditions are much more similar.

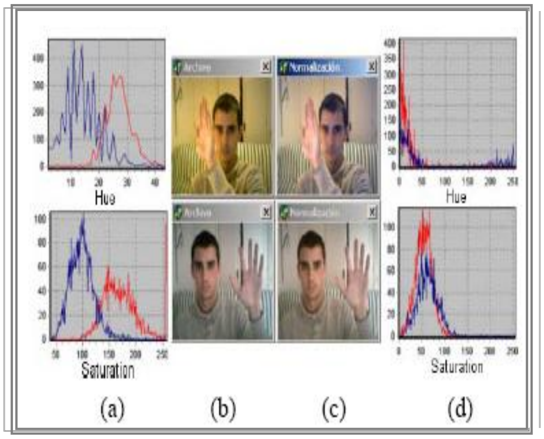

Figure 3. Skin detection: (a) Histogram for hue and saturation components before normalization operation, red line for artificial light and blue line for natural light; (b) original image under artificial and natural light conditions; (c) normalized image; (d) histogram for hue and
saturation components 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. 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 color and background. These ranges are defined by two rectangles in the HS plane: the *R1* rectangle for natural light ($0 \leq H \leq 15$; $20 \leq S \leq 120$) and the *R2* rectangle for artificial light ($0 \leq H \leq 30$; $60 \leq S \leq 160$). 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 neighborhoods 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 neighborhoods. 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 neighborhood. Optimal edge maps, where no redundant pixels could be found, were produced with the use of a 4 connectivity neighborhood.

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 center 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

The three-stage recognition has been proposed Kalman filtering process (level1) hidden Markov models (level 2) and graph matching (level 3):

Level 1

At this level the beginning image of a gesture is projected into the eigenspace made at the first level of the training phase, which maps onto a point. By finding the Euclidean distance of this point to all the representatives in this space a list of the representatives is formed, which is sorted in ascending order based on the Euclidean distances. The nearest representative at the top of the list represents a group of gestures that start with the same shape as the unknown gesture. However, because of variation in the position, the angle of hand and, of course, because of noise, this is not the best estimate and one should consider more than one representative. By taking α representative from the top of the list a group of gestures starting with one of the selected shapes is passed to the next level.

Level 2

By projecting the input gesture into the second common eigen space formed in the second level of the training phase the nearest sequence of symbols is extracted. The trained HMMs of the gestures forwarded from the first level are employed to calculate the likelihood of the extracted sequence. One can consider the HMM that results in the largest likelihood as the best match. However, because of the similarity of the gestures many gestures may have the same sequence of extracted code vectors (symbols) in some parts. Also, the large number of gestures makes the extracted sequence of code vectors very similar and a small amount of noise can change the extracted sequence of code vectors. Therefore, the gestures are sorted based on their likelihood at this stage. The correct gesture has either the highest likelihood or a small deviation from the highest one. So, a well chosen margin gives a few gestures with a small deviation from the greatest likelihood. These gestures are chosen to be compared carefully with the unknown gesture.

Level 3

By projecting the unknown gesture into the eigen spaces of the selected gestures we try to find the best match. The projections of the gesture in the subspaces form the individual manifolds. In each manifold is estimated by a graph. Therefore, there are two graphs in every subspace.

5. CONCLUSIONS AND FUTURE WORK

A fast processing process and a robust matching carried out through this 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 behavior 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 [14] and into virtual environment applications.

6. REFERENCES:

[1] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *PAMI*, 19(7):677–695, July 1997.

[2] W. Freeman. Computer vision for television and games. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 118, 1999.

[3] J. Triesch and C. vonder Malsburg. Robotic gesture recognition. In *Gesture Workshop*, pages 233–244, 1997.

[4] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *SCV95*, pages 265–270, 1995.

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

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

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

[8] R. O. Cornett, "Cued Speech", American Annals of the Deaf, 112:3-13, 1967.

[9] 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.

[10] 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.

[11] J. B. Kruskall and M. Liberman. The symmetric time warping algorithm: From continuous to discrete. In *Time Warps*. Addison-Wesley, 1983.

[12] L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice Hall, 1993.

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

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