Color Based Hand and Finger Detection Technology for User Interaction

Sung Kwan Kang, Mi Young Nam , Phill Kyu Rhee
Dept. of Computer Science & Engineering, Inha University
253, Yong-Hyun Dong, Nam-Gu
Incheon, South Korea
kskk1111@empal.com, rera@im.inha.ac.kr, pkrhee@inha.ac.kr

Abstract

The aim of this paper is to present the methodology for hand detection and propose the finger detection method. The detected hand and finger can be used to implement the non-contact mouse. This technology can be used to control the home devices such as curtain and television. Skin color is used to segment the hand region from background and counter is extracted from the segmented hand. Analysis of counter gives us the location of finger tip in the hand. We have performed extensive experiment and achieve very encouraging result.

1. Introduction

In recent years, there has been a tremendous amount of research on hand gesture recognition. Some of the earlier gesture recognition systems attempted to identify gestures using glove-based devices that would measure the position and joint angles of the hand [4]. However, these devices are very cumbersome and usually have many cables connected to a computer. This has brought forth the motivation of using no intrusive, vision-based approaches for recognizing gestures.

Gestures, particularly in sign language, involve significant motion of the hands. Thus, in developing a sign language recognition system, it is important to model both the motion (temporal characteristics) and shape (spatial characteristics) of the hand. While modeling the motion of the hand is imperative for sign language recognition, it is out of the scope of this thesis and is left for future work. Since only the spatial characteristics of the hand are of concern, temporal modeling of the hand will not be described here.

Discussions on temporal hand modeling can be found in [4] and [8]. Spatial modeling can be divided into two major categories: 3-D model-based approaches and appearance based or view-based techniques, as shown in the following figure. The following sections describe these two model types in greater detail.

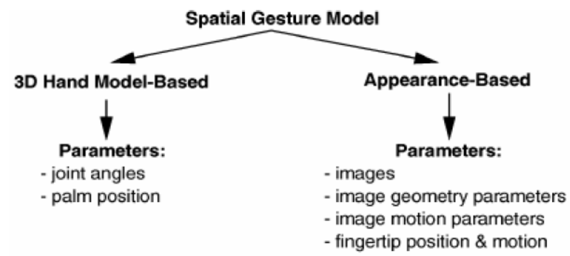


Figure 1. The kind of spatial gesture model

Three-dimensional models attempt to infer the 3-D pose of the hand. 3-D hand models are classified into two major groups: volumetric models and skeletal models.

Volumetric Models

Volumetric models aim to describe the 3-D appearance of the hand, as it would appear in real-life. They are commonly used in computer animation but have been recently used in computer vision applications [4]. Volumetric models are employed in vision-based hand gesture recognition by the approach of *analysis-by-synthesis*. *Analysis-by-synthesis* estimates the hand's posture by synthesizing the 3-D model of the hand, and then varying its parameters until the projection of the model on the image plane and the real hand image appears as the same visual image [4], [8]. Some volumetric models represent the surface of the hand with B-splines [10]. These are most popular in the field of computer animation since they are quite realistic [4].

However, these are too complex to be rendered in real-time gesture recognition applications. An alternative to these is in the use of geometric shapes such as cylinders, spheres, ellipsoids, and hyper rectangles to approximate the parts of the hand. These parts can then be combined to model the entire hand.

978-0-7695-3328-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICHIT.2008.292

Several systems [10], [11], [12], [23] have been proposed which use the approach of analysis-by-synthesis. Among these, the fastest frame rate achieved is 27 Hz.

While these models give a fairly realistic representation of the hand, they require many parameters. Obtaining these parameters with computer vision based techniques can be quite complex and time-consuming, which generally restricts these types of models from real-time use. In addition, these hand models are user dependent, since the model should be calibrated for each user, and thus they can only give approximate estimations [8].

Skeletal Models

Skeletal models represent the 3-D structure of the hand. While volumetric models require many parameters to accurately represent the actual appearance of the hand, skeletal models use a greatly reduced set of parameters to describe the structure of the hand. In order to understand skeletal models, it is first important to understand the structure of the human hand, shown in Figure 2.

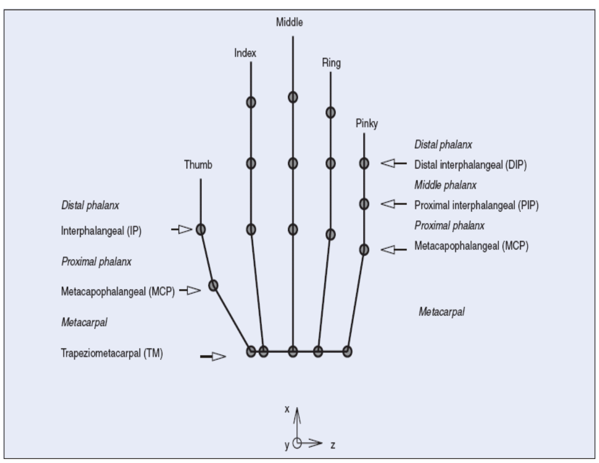


Figure 2. Skeletal structure of the human hand

The human hand consists of 5 fingers, each of which contains three joints. Except for the thumb, there are two degrees of freedom for metacarpophalangeal (MCP) joints and one degree of freedom for proximal interphalangeal (PIP) joints and distal interphalangeal (DIP) joints [8]. Taking into account all degrees of freedom for each joint and also considering global hand pose, the human hand has roughly 27 degrees of freedom [7], [8]. In skeletal models, each finger is represented as a kinematics chain where the palm is its base reference frame, the fingertips are the end-effectors, and inverse kinematics is involved in computing the joint angles [8]. However, a unique solution to the inverse kinematics problem cannot be guaranteed [8] and computation is rather complex.

Several systems have been proposed which use skeletal models [24], [25], [26]. Among systems that papered their operating rates, frame rates ranged between 8 fps and .02 fps (45 minutes per frame). Due to the computational complexity, skeletal models are not suited well for real-time gesture recognition applications.

The second major type of hand models is known as appearance-based models. Appearance-based models are derived directly from the information contained in the images. There are a variety of appearance-based models: those based on deformable templates, those that use hand image sequences, and those that use other image features such as shape representation features and image eigenvectors. Some appearance-based models are based on deformable templates. Deformable templates are the set of points on the outline or region of an object, used for interpolation to approximate the outline or region of an object [7]. Deformable templates consist of internal and external parameters. Internal parameters consist of an average set of points that describes the shape along with variability parameters that allow the shape to be deformed. External parameters are used to describe the global motion of the hand, which is generally described with rotations and translations. Two such systems which incorporate deformable models include [27] and [28]. In [28], a deformable template is used for hand tracking. However, the system suffers from scale and rotation confusion and "implausible model shapes" [28]. In [27] the hand shape is represented with a 3-D deformable template (Point Distribution Model). The results averaged angle differences of the 3-D hand posture of about 10 . – 20 , which is sufficiently accurate for sign language recognition [27]. However, the system requires that the model and the hand in the image overlap [27]. The method has also not been applied to sign language recognition.

Some appearance-based models are based on hand image sequences. Gestures, with these types of models, are depicted by a sequence of images themselves [7]. Motion history images (MHIs), images formed by the combination of the motion of pixels in a series of images over time, are one example. While systems that use these types of models may be good for small gesture sets, it would not be feasible to use such a model for a system such as sign language recognition that has a very large gesture set. Most appearance-based models are based on parameters of the hand image. Models under this category use parameters such as shape contours, image moments, and image eigenvectors. The position of fingertips in the image has also been used to help. distinguish between hand gestures. These types of parameters have been widely used for sign language recognition. Some examples of systems that use these types of appearance-based parameters are described in the following section.

The rest of the paper is organized as follows. Section 2 briefly introduces Vision based Recognition System. Section 3 describe about methodology proposing in this paper**.** Section 4 shows result that experiment using method proposing in this paper. Concluding remarks are given in Section 5.

2. Vision based Recognition System

Vision-based approaches involve using one or more video cameras to capture a person gesturing and using computer vision techniques to interpret each particular gesture. A vision-based gesture recognition system can be broken down into three main components: hand gesture modeling, hand gesture analysis, and hand gesture recognition. The gesture model describes how the hand gesture is to be represented. The type of application desired has a significant impact on the type of model that must be chosen. If an application with only a small number of gestures is needed, then a simple model can be used. However, in an application with a large gesture set, such as sign language recognition, a more detailed model will be required. However, there are trade-offs in choosing a more complex model as will be discussed later. After choosing a model, analysis is performed to compute the model parameters from the image features that are extracted from the video input streams. The analysis stage is followed by the recognition phase, which classifies the model parameters, representative of a specific gesture, while taking into consideration the model and in some cases grammar.

The following figure shows a diagram of a generic vision based gesture recognition system.

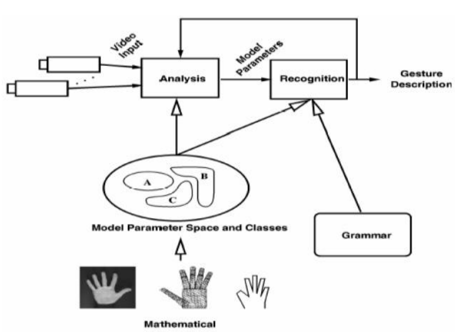


Figure 3. Vision-based gesture recognition system

The following sections describe the three main components of a vision-based gesture recognition system in more detail.

Feature Detection

The main task in the analysis stage is to detect relevant image features to be used to compute the model parameters. Before these hand features can be obtained, the hand in the image must first be detected and located in a process called localization or segmentation. After the hand region is detected, various features are computed, which are then used to determine the model parameters.

Hand Localization

The localization of the hand in the image can be done in several ways or in a combination of ways. One of the more popular ways of localizing the hand in the image is to use color segmentation. Because of the characteristic color of human hand, it can often be used solely to segment the hands in images. In the cases where gloves are worn, this color segmentation is trivial and very effective. There are many different techniques used to perform color segmentation including color histogram matching [36], look-up tables, and the use of probabilistic models. Alternatively, the motion of a user’s hand in images can be used to help detect the hand region from images. Several systems user standard background subtraction [38] and other techniques [36], [37] to detect areas of motion in images in order to locate the region of the hands. Color segmentation, motion, nd other visual indicators can be combined in another approach known as *fusion*. By combining these techniques, a more robust hand localization technique can be developed. n [36] color histogram segmentation and motion information is fused to locate the hands in images.

Feature Extraction and Parameter Computation

After locating the hands in the images, it is necessary to extract the required features to be used for computing the model parameters. Although various models have different parameters, the same feature can sometimes be used to compute several parameters. Some common features extracted include hand silhouettes [12], [41], contours [34], key points distributed along hand (fingertips, joints, palm) [25], [28], [33], [38], and distance-transformed images [40]. Silhouettes have been used as features for parameter computation in both 3-D based models [12], [41] and appearance based models. In [41], the silhouette features from multiple viewpoints are combined to construct a “voxel odel” which is then used to estimate the hand's joint angles. Contours can also be used to compute both 3-D model-based parameters and appearance based model parameters. Image-contour to model-contour matching is another way contours can be used in 3-D model-based approaches [7]. In appearance-based approaches, contours are often used to compute shape signatures. Key hand point features are also used in appearance-based models and 3-D based models. In [41], a 3-D model-based approach, eight key hand points (5 fingertips, wrist, middle finger joint, and thumb joint) are used to estimate the hand's joint angles. In [29], an appearance-based approach, the locations of colored markers located at the fingertips, palm, and back of hand are used to compute fingertip and hand distance parameters. In [42], a distance transformed image is used to compute palm size and hand direction parameters. In some appearance-based model approaches, the features extracted are treated as the actual hand parameters used in the recognition stage. For example, in [34], the hand contour feature is used directly as a parameter for the recognition phase.

3. Proposed Methodology

In this paper we propose a real time non-invasive hand tracking and finger recognition system. In this section we explain our method divided in three main steps. First step hand segmentation where the image region that contains the hand has to be located. In order to make this process it is possible to use shapes, but they can be changed greatly in interval that hand moves naturally[16]. So, we select skin-color to get characteristic of hand. The skin-color is a distinctive cue of hands and it is invariant to scale and rotation. In the next step we use the estimated hand state to extract several hand features to define a deterministic process of finger recognition.

Color cue is the main information that is being exploited to detect hand and thus finger tip. The method is based in a color model of the skin-colour pixels. After the hand is segmented from the background, a counter is extracted. The counter vector contains the series of coordinates of edges of hand. Then the processing of counter vector gives the location of the finger tip.

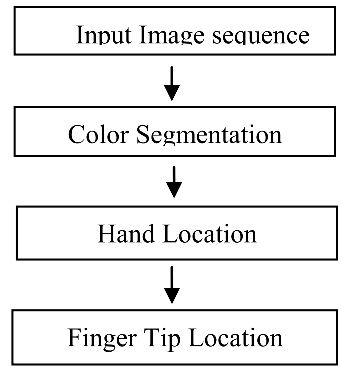


Figure 4. Finger tip location

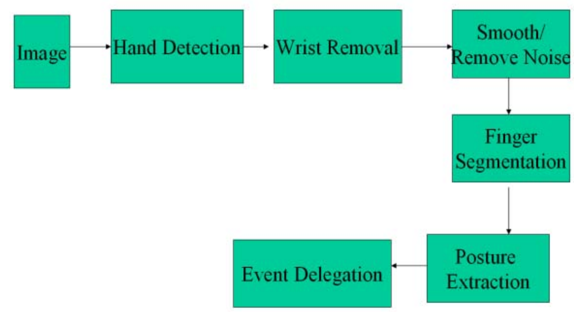
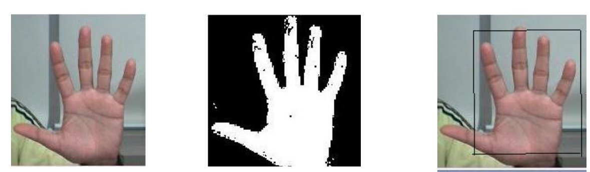


Figure 5. Flowchart for finger gesture recognition

3.1 Color Segmentation

Color Segmentation is done by analyzing the skin color over the skin color range. The simple model of skin color indicates that red component of the skin color is in the range of 37 and 60, whereas the green component of the skin color is between 28 and 34.



Original Image Segmented Hand Located Hand

Figure 6. Skin color segmentation of the human hand

RGB color space is most native to many video capture devices. However, RGB color space is very sensitive to change in brightness and intensity. HSI and YUV color spaces are often used, as the intensity component can be treated separately to the chrominance components. The transformation is done by software conversion, which can be quite expensive. Price et al. (2000) have proposed a modified HSI by extending 360 degrees of hue for faster processing. For this project, we use YUV for less expensive conversion from RGB. We use the OpenCV implementation (OpenCV 2005) which takes advantage of MMX instruction to perform faster RGB to YUV conversion [3].

3.2 Contour Extraction

Contour Extraction is the process of retrieving the location of pixel value of the edges of the detected objects from an image. Once the hand blob is detected then, the edge vector of the particular blob can be retrieved using Edges () function of Blob class. The figure below shows the contour of hand plotted in original image.

Refined contours can be obtained by minimizing the snake energy of each of the returned contours with respect to the binary image; this work is also adopted in work [3]. Finally, we obtain the blob representation of the hand applying a connected components algorithm to the probability image, which groups pixels into the same blob.

Figure 7. Contour extracted from located hand

3.3. Contour Analysis and Finger Detection

Once we have contour vector, i.e. the silhouette of the hand region, the vector can be analyzed to find the finger location. In the current version of application, only one finger whose height is maximum among others is found by contour analysis. The contour vector contains the x and y coordinates of the each point of hand silhouette. First maximum value of y coordinate is searched and the corresponding x coordinate corresponding to y coordinate is used to locate the finger tip.

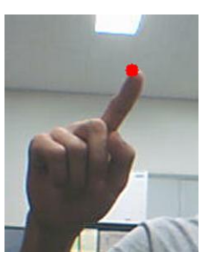
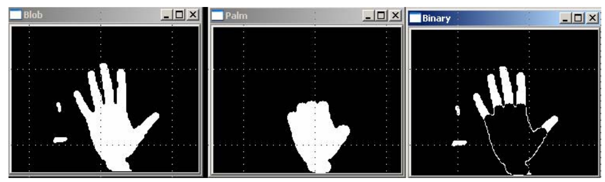


Figure 8. Location of finger-tip

Application of finger detection can be in visual user interface where human hand is a pointing device. Once the finger tip is located; the position of finger tip can be calibrated as a mouse position. As a use moves his hand, in the plane that is parallel to camera, then mouse curser can be moved. However to implement the mouse click operation, two hands are needed. One possible approach is dedicate one hand for a cursor position and other hand for mouse right and left clicks functionality.



a) Hand Object b) after Morphological Opening c) A-B

Figure 9. Using Morphological Operation to detect finger blobs

To find the fingers blob in an image, morphological opening can be used. As shown in figure, image A is a hand blob obtained by color segmentation. Image B is a palm segmented from the hand and is retrieved by applying the morphological opening operation with elliptical structuring element. By subtracting the image b from image a, fingers can be detected in image.

3.4. Finger Recognition

Our finger number consists of five finger and twelve finger shapes in order to fulfill the application's requirements. The finger number correspond to a fully opened hand with separated fingers, an opened hand with fingers together, in part or completely, in the camera's field of view. These gestures also can express *Enter* and *Backspace* gesture in addition to numbers. When we express these gestures and is recognized by relevant number, hardly be influenced on distance of camera and hand. Finally, the valid finger gesture transitions that the user can carry out are defined in Fig.1.

The process of finger gesture recognition starts when the user's hand is placed in front of the camera field of view and the fingers are in gesture to be predefined, that is, the hand fully opened with separated fingers. For avoiding fast finger gesture changes that were not intended, every change should be kept fixed for 5 frames approximately, if not the finger gesture does not change from the previous recognized gesture.

- **Number representation by hand**

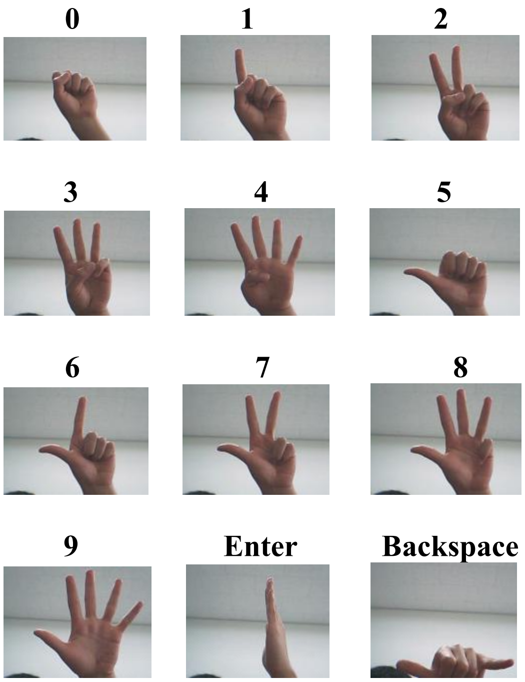


Figure 10. Valid finger number

Figure11 captured image that recognize several finger gestures in application that propose in this paper. We can see recognize exactly number that is predefined if fingers interval is separated clearly in this image. Figure 12 shows that express multiplication result using finger recognition in proposed system.

Figure 11. Proposed finger recognition system

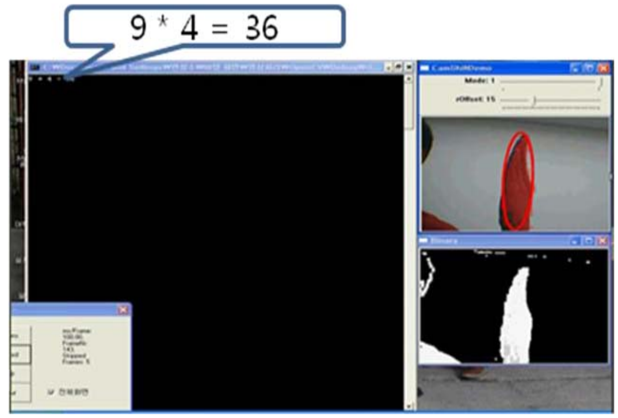


Figure12. Expression of Calculator result in proposed system

4. Experiment Results

In this section we describe the accuracy of our hand tracking and gesture recognition algorithm. The application has been implemented in Visual C++ using proposed methodology and the OpenCV libraries. The application has been tested on a Pentium IV running at 2.40 GHz. The images have been captured using a Logitech Messenger WebCam with USB connection. The camera provides 640x480 images at a capture and processing rate of 30 frames per second. For the performance evaluation of the hand detection and gesture recognition, the system has been tested 6 times respectively on a set of 5 users. Each user has performed a predefined set of 6 movements and therefore we have 360 gestures to evaluate the application results. It is natural to think that the system's accuracy will be measured controlling the performance of the desired user movements for managing the calculator. This sequence included all the application possible states and transitions. Figure 13 shows the performance evaluation results. These results are represented using a bidimensional matrix with the application states as columns and the number of appearances of the gesture as rows. The columns are paired for each gesture: the first column is the number of tests of the gesture that has been correctly identified; the second column is the total number of times that the gesture has been carried out. As it can be seen in Fig. 13, the finger recognition gesture works fine for a 99% of the cases.

Result Sample

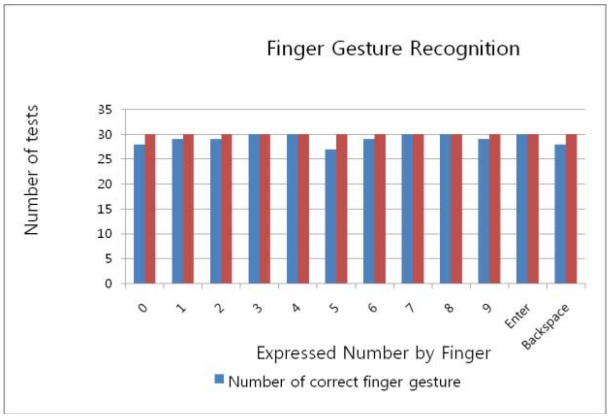


Figure 13. System's performance evaluation results

5. Conclusion

A color based method for hand location is presented and hand contour analysis is done to locate the finger tip. Color detection algorithm is simple and more robust method like probabilistic method of color modeling and HSV modeling is suggested for future extension. Once the hand counter is extracted from hand blob, highest position of contour is considered as finger tip. In future the more method of contour analysis can be done to locate all five fingers .This will give more flexibility to interpret the gestures. Furthermore, hand detection method using texture and shape information can be used to maximize the accuracy of detection in cluttered background. Similar method adapted by voila and jhone to detect faces in image can be used to detect fixed shaped hand however deformable and spring model matching can be incorporated to detect the deformable structure of hand.

References

[1] C. Vogler and D. Metaxas, "A Framework for Recognizing the Simultaneous Aspects of American Sign Language," *Computer Vision and Image Understanding, vol.* 81, pp.358-384, 2001.

[2] K. G. Derpanis, "A Review of Vision-Based Hand Gestures," Unpublished. Feb.2004

[3] C. Vogler and D. Metaxas, " Towards scalability in ASL Recognition: Breaking Down Signs into Phonemes," in *Gesture Workshop*, pp. 17-99, 1999.

[4] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review." *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 19 issue 7, pp 677-695, July 1997.

[5] S. Gunn, " Support Vector Machines for Classification and Regression," ISIS Tech. Paper, University of Southampton, 1998.

[6] A. Khotanzad and Y. H. Hong, " Invariant Image Recognition by Zernike Moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12 issue 5, pp. 489-497, May 1990.

[7] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," In *Gesture Workshop*, pp. 103–115, GW 99, France. March 1999.

[8] Y. Wu and T.S. Huang, "Hand modeling, analysis and recognition," *IEEE Signal Processing Magazine*, vol. 18 issue 3, pp. 51-60, May 2001.

[10] J.J. Kuch and T.S. Huang, "Vision-based hand modeling and tracking for virtual teleconferencing and telecollaboration," in *Proc. IEEE Int. Conf. Computer Vision*, Cambridge, MA, pp. 666-671. June 1995

[11] J. Davis and M. Shah, "Visual gesture recognition," *Vision, Image, and Signal Processing*, vol. 141, pp. 101-106, Apr. 1994.

[12] N. Shimada et al., "Hand gesture estimation and model refinement using monocular camera—Ambiguity limitation by inequality constraints," in *Proc. 3rd Conf. Face and Gesture Recognition*, pp. 268-273, 1998.

[13] J. Rehg and T. Kanade, "DigitEyes: vision-based human hand tracking," School of Computer Science Technical Paper CMU-CS-93-220, Carnegie Mellon Univ., Dec.1993.

[14] "OpenCV (Open Source Computer Vision Library)," Intel Corporation. 2 Aug. 2004 Available at: http://www.intel.com/research/mrl/research/opencv/.

[15] T. Starner and A. Pentland, "Real-time american sign language recognition using desk and wearable computer based video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1371–1375, December 1998.

[16] T. Heap, D. Hogg, "Wormholes in shape space: tracking through discontinuous changes in shape", *Proc. Sixth International Conference on Computer Vision, ICCV98*, pp. 344-349, 1998.

[17] N. Janibata and N. Shimada, " Extraction of Hand Features for Recognition of Sign Language Words," *The 15th International Conference on Vision Interface,* Calgary, Canada*,* May 27-29, 2002.

[18] C. Nölker, H. Ritter, "Detection of Fingertips in Human Hand Movement Sequences," *Gesture and Sign Language in Human-Computer Interaction*, I. Wachsmuth and M. FroÈhlich, eds., pp. 209-218, 1997.

[19] C. W. Ng and S. Ranganath, "Real-time gesture recognition system and application," *Image and Vision Computing,* vol. 20, issues 13-14, pp. 993-1007, 1 December 2002.

[20] K. Abe, H. Saito, and S. Ozawa, "Virtual 3D interface system via hand motion recognition from two cameras," *IEEE Transactions on Systems, Man and Cybernetics*, Part A, vol. 32 , issue 4 , pp. 536 – 540, July 2002.

[21] D. Zhang, " Image Retrieval Based on Shape," PhD Thesis. Monash University, Australia, March 2003.

[22] T. Pavlidis. Algorithms for Graphics and Image Processing. Computer Science Press, pp.143, 1982.

[23] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice Hall, Upper Saddle River, NJ. 2001.
[24] N. Shimada, K. Kimura and Y. Shirai, "Real-time 3D Hand Posture Estimation based on 2-D Appearance Retrieval Using Monocular Camera," *Proc. Int. WS on RATFG-RTS*, pp. 23-30, 2001.
[25] J. Lee and T. Kunii, "Model-based analysis of hand posture," *IEEE Comput. Graph. Appl*., vol. 15, no. 5, pp. 77-86, Sept. 1995.
[26] Y. Wu and T.S. Huang, "Capturing articulated human hand motion: A divide-andconquer approach," in *Proc. IEEE Int. Conf. Computer Vision*, Corfu, Greece, Sept. 1999, pp. 606-611.
[27] K. Kwon, H. Zhang, F. Dornaika, "Hand Pose Recovery with a Single Video Camera" , *in Proc. of ICRA 2001*, Seoul, Korea, May 21-26, 2001.
[28] H. Fillbrandt, S. Akyol, K. F. Kraiss, " Extraction of 3D Hand Shape and Posture from Image Sequences for Sign Language Recognition." *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, vol. 17, pp. 181-186, October 2003.
[29] T. Heap and D. Hogg, " Towards 3D hand tracking using a deformable model," *in Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition*, Killington, VT, pp. 140-145, 1996.
[30] B. Bauer and H. Hienz, *"*Relevant features for video-based continuous sign language recognition," *in Proc. of Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 440-445, March 2000.
[31] Y. Shirai, N. Tanibata, N. Shimada, *"*Extraction of hand features for recognition of sign language words*,"* VI'2002, *Computer-Controlled Mechanical Systems*, Graduate School of Engineering, Osaka University, 2002.
[32] K. Grobel and M. Assam, " Isolated sign language recognition using hidden Markov models," *in Proc. of the IEEE International Conference on Computational Cybernetics and Simulation*, Orlando, pp. 162-167, 1997.
[33] T. Starner, A. Pentland , J. Weaver, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20 issue 12, pp. 1371-1375, December 1998.
[34] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," *in* P*roc.of International Symposium on Computer Vision*, pp. 265–270, 21-23 Nov. 1995.
[35] Y. Hamada, N. Shimada and Y. Shirai, "Hand Shape Estimation under Complex Backgrounds for Sign Language Recognition" , *in Proc. of 6th Int. Conf. on Automatic Face and Gesture Recognition,* pp. 589-594, May 2004.
[36] J. Isaacs and J. S. Foo, "Hand pose estimation for American sign language recognition," *in Proc. of the Thirty-Sixth Southeastern Symposium on System Theory*, pp. 132 – 136, March 14-16, 2004
[37] M. Frigola, J. Fernandez, and J. Aranda, "Visual human machine interface by gestures**,"** *in Proc. IEEE International Conference on Robotics and Automation*, vol. 1, pp. 386 – 391, Sept. 14-19 2003.
[38] K. Imagawa, S. Lu, and S. Igi, "Color-based hands tracking system for sign language recognition" , *in Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 462 – 467, April 14-16, 1998.
[39] C. Graetzel, S. Grange, T. Fong, and C. Baur, "A Non-Contact Mouse for Surgeon- Computer Interaction," *IEEE Medical Image Computing and Computer Assisted Intervention, Toronto, Canada interface*, November 2003.
[40] C.-H. Teh and R. T. Chin, "On image analysis by the methods of moments" , *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, no. 4, pp. 496 – 513, July 1988.
[41] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Brooks/Cole Publishing Company, 1999.
[42] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, "A hand-pose estimation for vision-based human interfaces*,"* *IEEE Transactions on Industrial Electronics*, vol. 50, issue 4 , pp. 676–684, Aug. 2003.
[43] J. Lin, Y. Wu, and T. S. Huang, "Capturing human hand motion in image sequences," *in Proc. Workshop on Motion and Video Computing*, pp. 99-104, Dec. 5-6, 2002.