International Journal of Electronics and Electrical Engineering
ISSN : 2277-7040 Volume 2 Issue 3 (March 2012)
http://www.ijecee.com/ https://sites.google.com/site/ijeceejournal/

REAL TIME HAND GESTURE RECOGNITION USING SIFT

Pallavi Gurjal (pallavigurjal@yahoo.co.in), Kiran Kunnur(k.kunnur@gmail.com).

Abstract. The objective of the gesture recognition is to identify and distinguish the human gestures and utilizes these identified gestures for applications in specific domain. In this paper we propose a new approach to build a real time system to identify the standard gesture given by American Sign Language, or ASL, the dominant sign language of Deaf Americans, including deaf communities in the United States, in the English-speaking parts of Canada, and in some regions of Mexico. We propose a new method of improvised scale invariant feature transform (SIFT) and use the same to extract the features. The objective of the paper is to decode a gesture video into the appropriate alphabets.

Keywords: Scale invariant feature, gesture recognition, template matching, frame set

1 Introduction

In machine learning, pattern recognition is the assignment of some sort of output value (or label) to a given input value (or instance), according to some specific algorithm. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to do "fuzzy" matching of inputs. This is opposed to pattern matching algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern

recognition, pattern matching is generally not considered a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing similar-quality output to the sort provided by pattern-recognition algorithms.

Gestures are a powerful means of communication among humans. In fact gesturing is so deeply rooted in our communication that people often continue gesturing when speaking on the telephone. Hand gestures provide a separate complementary modality to speech for expressing ones ideas. Information associated with hand gestures in a conversation is degree, discourse structure, spatial and temporal structure.

The remarkable ability of the human vision is the gesture recognition, it is noticeable mainly in deaf people when they communicating with each other via sign language and with hearing people as well. In this paper we take up one of the social challenges to give this set of mass a permanent solution in communicating with normal human beings. In this paper we consider the ASL gestures as the images to be extracted from the gesture video and further decode it with English alphabet. American Sign Language, or ASL, for a time also called Ameslan, is the dominant sign language of Deaf Americans, including deaf communities in the United States, in the English-speaking parts of Canada, and in some regions of Mexico. [2] Although the United Kingdom and the United States share English as a common language, British Sign Language (BSL) is quite unlike ASL, and the two languages are not mutually intelligible. ASL is instead related to French Sign Language.

Besides North America, ASL is also used, sometimes alongside indigenous sign languages, in the Philippines, Malaysia, Singapore, the Dominican Republic, El Salvador, Haiti, Puerto Rico, Côte d'Ivoire, Burkina Faso, Ghana, Togo, Benin, Nigeria, Chad, Gabon, the Democratic Republic of the Congo, the Central African Republic, Mauritania, Kenya, Madagascar, Zimbabwe, Barbados, Bolivia, China, Mauritania, and Jamaica. Like other sign languages, its grammar and syntax are distinct from any spoken language, including English.

In ASL, finger spelling in the American manual alphabet is used primarily for spelling out names or English terms which do not have established signs. However, it is also used for emphasis (for example, finger spelling #YES is more emphatic than signing YES), for clarity, and for instruction.

Besides finger spelling entire English words, ASL frequently incorporates the first letter of an English word into an equivalent ASL sign to distinguish related English concepts which do not otherwise have separate signs. For example, two hands tracing a circle is the sign for a group of people. Several kinds of groups can be specified by hand shape: when made with a 'C' hand shape, the sign means class, when made with an 'F' hand shape, family. Such signs are called initialized signs. Only a small number of common signs are disambiguated through initialization in this way. However, a number of other signs incorporate a letter from English. The sign for elevator, for example, is made by moving an 'E' hand shape up and down the extended index finger of the other hand, but there is no related sign with a different letter. Fig (1) shows the gestures for the English alphabets [2] these guidelines as closely as possible.

The fundamental principle applied for gesture recognition is that of pattern recognition. In pattern recognition, direct correlation of the images would consume

ample amount of time, as the size of images are too large and the number of comparisons are many. In order to rectify the problem of time consumption, a new approach was Feature extraction followed by pattern recognition using the features. In this paper scale invariant feature transform is used to extract the features from the image.

The idea of using local oriented features over different visual scales has shown itself to be perhaps the most effective visual primitive for object recognition, robot localization, video retrieval and recently stitching images into panoramas. The first effective use of this idea was Christoph von der Malsburg's use of oriented Gabor filters over different scales linked in a graph. Recently, David Lowe has improved on the core idea by finding stable oriented features that indicate their scale (~depth) with his Scale Invariant Feature Transform (SIFT). In this paper we have attempted to give improved version of SIFT features first by simplifying the algorithm; then by improving the SIFT keys as judged by recognition accuracy in outdoor scenes. The rest of the paper is organized as follows: in section 1 A brief description of scale invariant feature transform is given. Section 2.The procedure involved in video file extraction is explained. In section 3 we describe the gesture recognition algorithm implemented along with the frame extraction from video. In section 4 we discuss the results and highlight future work. And finally, section 5 concludes this paper.

Figure [1] ASL Gestures

2 BRIEF DESCRIPTION OF SFIT

The SIFT algorithm takes an image and transforms it into a collection of local feature vectors. Each of these feature vectors is supposed to be distinctive and invariant to any scaling, rotation or translation of the image. In the original implementation, these features can be used to find distinctive objects in different images and the transform can be extended to match faces in images. This report describes our own implementation of the SIFT algorithm and highlights potential direction for future research. Scale-invariant feature transform (or SIFT) is an algorithm in computer vision to detect and describe local features in images. For any object in an image, interesting points on the object can be extracted to provide a "feature description" of the object. This description, extracted from a training image, can then be used to identify the object when attempting to locate the object in a test image containing many other objects. To perform reliable recognition, it is important that the features extracted from the training image are detectable even under changes in image scale, noise and illumination. Such points usually lie on high-contrast regions of the image, such as object edges. Another important characteristic of these features is that the relative positions between them in the original scene shouldn't change from one image to another. For example, if only the four corners of a door were used as features, they would work regardless of the door's position; but if points in the frame were also used, the recognition would fail if the door is opened or closed. Similarly, features located in articulated or flexible objects would typically not work if any change in their internal geometry happens between two images in the set being processed. However, in practice SIFT detects and uses a much larger number of features from the images, which reduces the contribution of the errors caused by these local variations in the average error of all feature matching errors.

The approach of SIFT feature detection taken in our implementation is similar with the one taken by Lowe et. [1], which is used for object recognition. Our method is implemented as the following stages: Creating the Difference of Gaussian Pyramid, Extreme Detection, key point Elimination, and Orientation Assignment.

2.1 Difference of Gaussian

To construct a Gaussian "scale space" function from the input image we go for convolution (filtering) of the original image with Gaussian functions of varying widths.

Where

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y). \quad (2)$$

In the above equation D(x, y, σ) is the difference of Gaussian calculated as the difference between two filtered images, one with k multiplied by scale of the other. [1]. I(x, y) is the input image. G(x, y, σ) is the Gaussian function given as follows;

$$G(x, y, \sigma) = \frac{1}{2\pi}\exp\{-(x^2 + y^2)/2\} \quad (3)$$

Each octave of scale space in divided into an integer number, s, of intervals and we let k=2^ (1/s) We produce s+3 images for each octave in order to form s+2 difference of Gaussian (DoG) images and have plus and minus one scale interval for each DoG for the extrema detection step. Once a complete octave has been processed, we subsample the Gaussian image that has twice the initial value of σ by taking every second pixel in each row and column. This greatly improves the efficiency of the algorithm at lower scales.

2.2 Extrema detection

This stage is to find the extrema points in the DOG pyramid. To detect the local maxima and minima of D(x, y, σ), each point is compared with the pixels of all its 26 neighbors. If this value is the minimum or maximum this point is an extrema. We then improve the localization of the key point to sub pixel accuracy, by using a second order Taylor series expansion. This gives the true extrema location as:

$$y = -\left(\frac{\partial^2 z}{\partial x^2}\right) \quad (4)$$

$$Z = \frac{1}{y}\frac{\partial D}{\partial x} \quad (5)$$

Where D and its derivatives are evaluated at the sample point and x=(x, y , σ)T is the offset from the sample point.

2.3 Key points Elimination

This stage attempts to eliminate some points from the candidate list of keypoints by finding those that have low contrast or are poorly localized on an edge.[1]. The value of the key point in the DoG pyramid at the extrema is given by:

$$D(z) = D + \frac{1}{2}\frac{\partial D^{-1}}{\partial x} z \quad (6)$$

To eliminate poorly localized extrema we use the fact that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. A 2x2 Hessian matrix, H, computed at the location and scale of the key point is used to find the curvature. With these formulas, the ratio of principal curvature can be checked efficiently.

$$H = \begin{matrix} D_{XX} & D_{XY} \\ D_{YX} & D_{YY} \end{matrix} \quad (7)$$

$$\frac{D_{XX} + D_{YY}}{D_{XX} D_{YY} - (D_{XY}^2)} < \frac{r+1^2}{r} \quad (8)$$

So if inequality (8) fails, the key point is removed from the candidate list

2.4 Orientation Assignment

This step aims to assign a consistent orientation to the keypoints based on local image properties. The gradient magnitude, m(x, y), and orientation, θ(x, y), are precomputed using pixel differences:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (9)$$

$$\theta(x,y) = \arctan((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))) \quad (10)$$

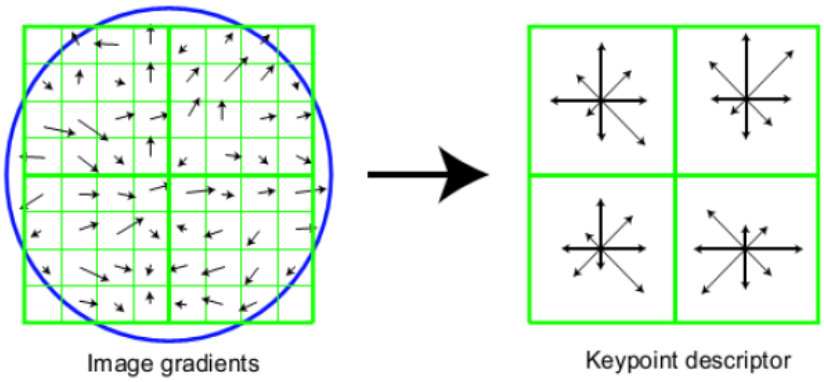

Figure [2] Orientation Assignment.

Each sample is weighted by its gradient magnitude and by a Gaussian-weighted circular window with σ that is 1.5 times that of the scale of the key point. Peaks in the orientation histogram correspond to dominant directions of local gradients. We locate the highest peak in the histogram and use this peak and any other local peak within 80% of the height of this peak to create a key point with that orientation. Some points will be assigned multiple orientations if there are multiple peaks of similar magnitude. A Gaussian distribution is fit to the 3 histogram values closest to each peak to interpolate the peaks position for better accuracy.
This computes the location, orientation and scale of SIFT features that have been found in the image. These features respond strongly to the corners and intensity gradients.

3 FRAME EXTRACTION

A film frame, or just frame, is one of the many single photographic images in a motion picture. The individual frames are separated by frame lines. Normally, 24 frames are needed for one second of film. In ordinary filming, the frames are photographed automatically, one after the other, in a movie camera. In special effects or animation filming, the frames are often shot one at a time. The term may also be used more generally as a noun or verb to refer to the edges of the image as seen in a camera viewfinder or projected on a screen. Thus, the camera operator can be said to keep a car in frame by panning with it as it speeds past. The size of a film frame varies, depending on the still film format or the motion picture film format. In the smallest 8 mm amateur format for motion pictures film, it is only about 4.8 by 3.5 mm, while an IMAX frame is as large as 69.6 by 48.5 mm. The larger the frame size is in relation to the size of the projection screen, the sharper the image will appear. Understanding these key features of a frame in video, extracting a frame from video is simple task.

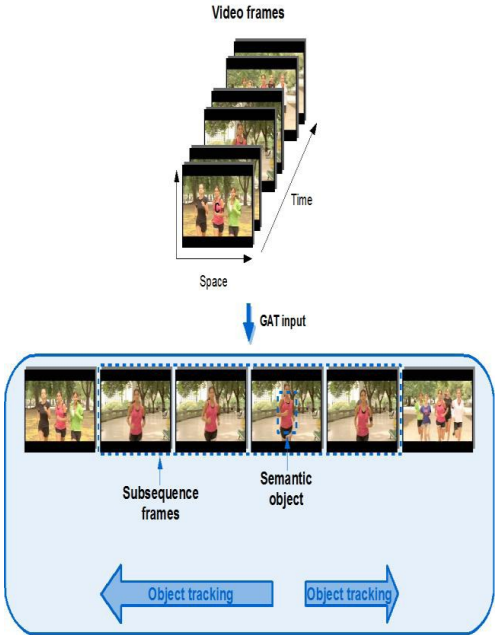

Figure [3] Frame Extraction.

4 ALGORITHM

The Algorithm is divided in four sub steps as follows

Step 1: Read the video in matlab with one caution that the video file is in AVI format.

PART [1] Extraction of the frame from the video

Step [2]: Obtain the number of frames in the video. With this the total time of the video gets divided and time of image is known.
Step [3] : Capture the object occurring in this time span and save in the folder as frame1
Step [4]: Loop through the movie, writing all frames out. Each frame will be in a separate file with unique name.

PART [2] Adjusting the size of each image

Step [5]: once the frame is obtained, the part of image which does not carry any information is selected.
Step [6]: This part of image is excluded by cropping the rest of the part of image.

PART [3]: Extract the Sift features of each frame

Step [7]: Difference of Gaussian is obtained using the equations mentioned above, and image pyramid is built.
Step [8]: scale space feature detector based upon difference of Gaussian filters and select features based upon their maximum response in scale space.

Parameters:

(1) Scaling factor between levels of the image pyramid
(2) Threshold value for maxima search (minimum filter response considered)
(3) Radius: radius for maxima comparison within current scale

(4) Radius2: radius for maxima comparison between neighboring scales
(4) Radius3: radius for edge rejection test
(5) Minimum separation for maxima selection.
(6) Maximum ratio of eigen values of feature curvature for edge rejection.
Step [9]: Key point elimination based on the condition (8) as mentioned above.
Step [10]: Orientations are assigned to features and using a function features are shown over the image

PART [4] Correlation of features for recognition
Step [11]: The features of database are loaded and the features of input image are correlated with that of database to find the English alphabet for the input frame.

5 RESULTS AND CONCLUSION

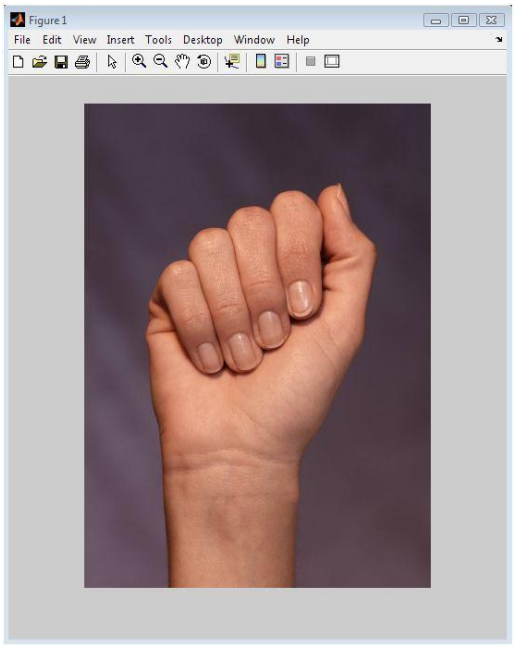

Figure [4] Input image

The above image is one of the input images. The sign is for the English alphabet A. In the above image some part of the image, i.e. the lower part does not carry any information hence in the first step we attempt to corp. this part of image.

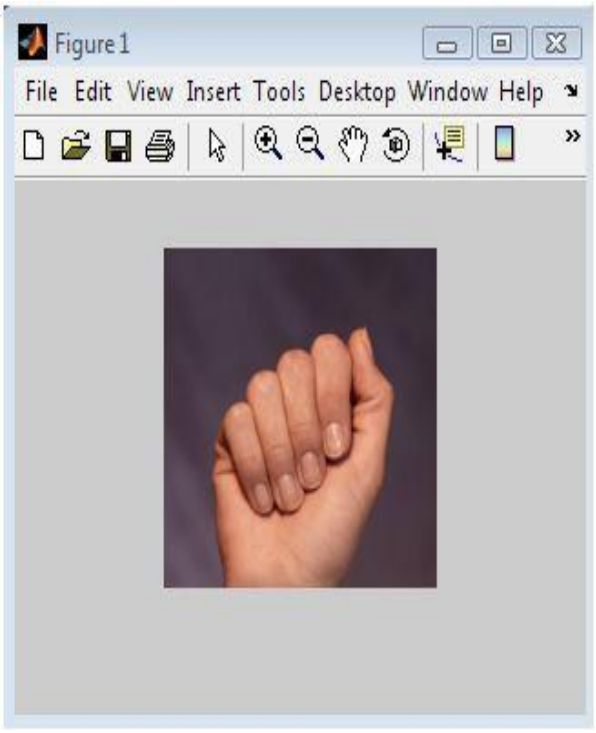

Figure [5] Image (after corp.)

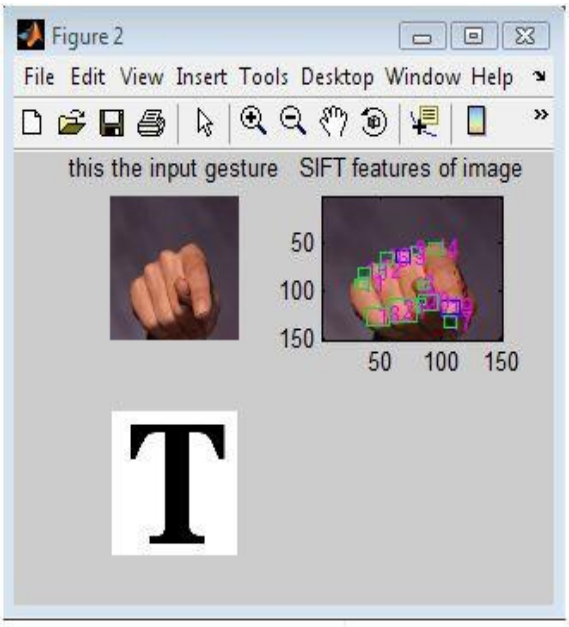

Figure [6] Correlated output for T

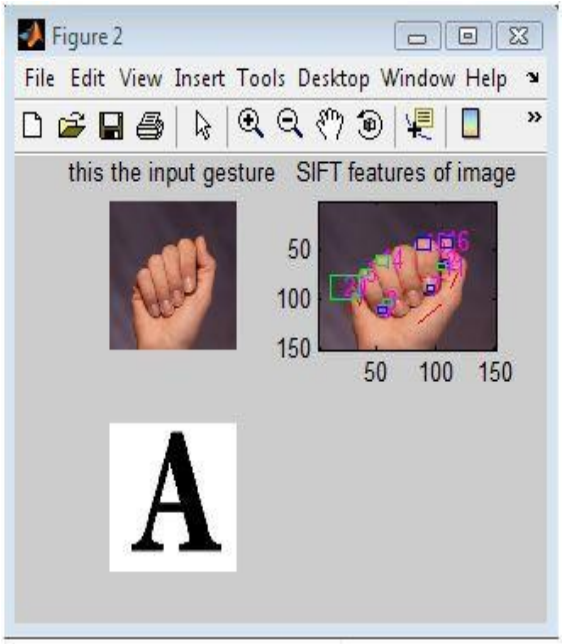

Figure [7] Correlated output for A

Figure [8] Frame Extraction

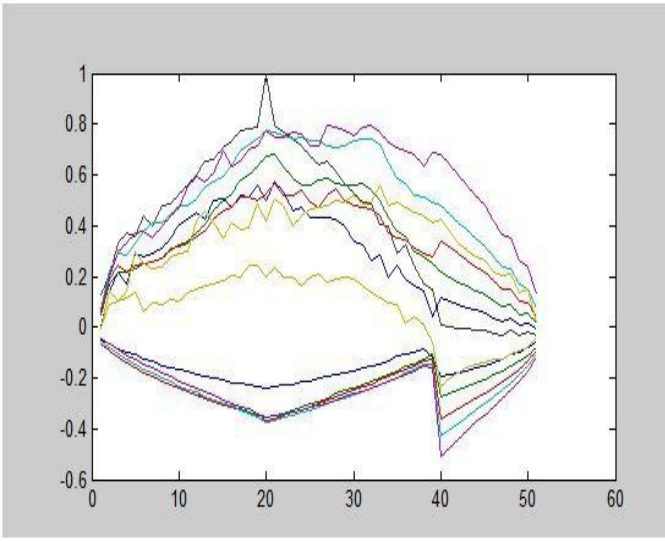

Figure [9] Correlation

6 Conclusion:

Figure (5): is the output obtained after adjusting the size of image. Figure (6) & (7) gesture of alphabets A & T being recognized. Figure (8) is the output of frame extraction from video. Figure (9) is the correlation output. In today's digitized world, processing speeds have increased dramatically, with computers being advanced to the levels where they can assist humans in complex tasks. Yet, input technologies seem to cause a major bottleneck in performing some of the tasks, under-utilizing the available resources and restricting the expressiveness of application use.

Hand Gesture recognition comes to rescue here. With our algorithm we were able to decode a video successfully with 40 frames. The frame extraction of 40 frames from video took about 40 seconds. The features were efficiently extracted using SIFT. The SIFT features described in our implementation are computed at the edges and they are invariant to image scaling, rotation, addition of noise. They are useful due to their distinctiveness, which enables the correct match for keypoints between faces.

References

[1]Lowe, D.G..2004. Distinctive Image Features from Scale-Invariant Keypoints. January 5, 2004
[2] YU MENG and Dr. Bernard Tiddeman(supervisor) Implementing the Scale Invariant Feature Transform(SIFT) Method.
[3] Matching Images with Different Resolutions.
[4] Mikolajczyk, K. & Schmid, C. 2001. Indexing based on scale invariant interest points. International Conference on Computer Vision, Vancouver, Canada (July 2001), pp. 525--531.
[5] Helmer, S.&Lowe,D.G..Object Class Recognition with Many Local Features.
[6] A. Mulder, "Hand gestures for HCI", Technical Report 96-1, vol. Simon Fraster University, 1996
[7] F. Quek, "Towards a Vision Based Hand Gesture Interface", pp. 17-31, in Proceedings of Virtual Reality Software and Technology, Singapore, 1994.
[8] Ying Wu, Thomas S Huang, " Vision based Gesture Recognition : A Review", Lecture Notes In Computer Science; Vol. 1739 , Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction, 1999
[9] K G Derpains, "A review of Vision-based Hand Gestures", Internal Report, Department of Computer Science. York University, February 2004
[10] Richard Watson,"A Survey of Gesture Recognition Techniques", Technical Report TCD-CS-93-11, Department of Computer Science, Trinity College Dublin, 1993.
[11] Y. Wu and T.S. Huang, "Hand modeling analysis and recognition for vision-based human computer interaction", IEEE Signal Processing Mag. – Special issue on Immersive Interactive Technology, vol.18, no.3, pp. 51-60, May 2001
[12] H. Zhou, T.S. Huang, "Tracking articulated hand motion with Eigen dynamics analysis", In Proc. Of International Conference on Computer Vision, Vol 2, pp. 1102-1109, 2003 .
[13] JM Rehg, T Kanade, "Visual tracking of high DOF articulatedbstructures: An application to human hand tracking" ,In Proc. European Conference on Computer Vision, 1994

[14] A. J. Heap and D. C. Hogg, "Towards 3-D hand tracking using a deformable model", In 2nd International Face and Gesture Recognition Conference, pages 140–145, Killington, USA,Oct. 1996.
[15] Y.Wu, L. J. Y., and T. S. Huang. "Capturing natural hand Articulation".In Proc. 8th Int. Conf. on Computer Vision, volume II, pages 426–432,Vancouver, Canada, July 2001.
[16] B. Stenger P. R. S. Mendonc¸a R. Cipolla, "Model-Based 3D Tracking of an Articulated Hand" , In proc. British Machine Vision Conference, volume I, Pages 63-72, Manchester, UK, September 2001