International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Hand Gesture Animation Model for Local Hand Motion

Bidit Hazarika[1], Dipankar Das[2], Dibyajyoti Changkakoti[3], Abhinov Deka[4], Adarsh Pradhan[5]

[1,2,3,4,5] Department of Computer Science and Engineering, GIMT, Guwahati, India

Available online at: www.ijcseonline.org

Accepted: 18/May/2018, Published: 31/May/2018

***Abstract*—**This paper aims to introduce a technique concept for facilitating communication between deaf & dumb and normal people. We propose a system which helps in translating speech/text into equivalent hand gestures. This can be done by image processing and/or animation techniques. With this application we hope to reduce the communication gap between _deaf & dumb‘ and normal people by eliminating the need of a human translator.

Keywords*—Hand gesture, Gesture Animation , Hand model, Finger pose estimation*

1 INTRODUCTION

Gestures are a form of nonverbal communication in which visible bodily actions are used to communicate important messages, either in place of speech or together and in parallel with spoken words . Gestures include movement of the hands , face , or other parts of the body . In this particular project , we specifically concentrate with hand gestures . Hand gestures are universally adopted means of communication to convey message in the form of sign language . Although the deaf and dumb are able to communicate using a mixture of hand gestures . Therefore , a normal human needs to have some knowledge about the sign language and should be able to make the sign language gestures in order to communicate with a deaf and dumb person . By understanding and animating hand gestures , we can facilitate communication between the normal person and the deaf and dumb . The whole communication process can be seen in which the computer should be able to recognize gestures and convert a particular sign to its corresponding message in the form of speech.

This will help a dumb person to convey message to a normal person . On the other hand, the system should be able to recognize any spoken word / sentence or text and create the corresponding sign (hand gesture) through animation . This will help a normal being to communicate with a deaf person

. So, the system will mainly act as an interpreter between a normal being deaf and a mute person . That is why, automatic interpretation and animation of hand gestures has become an important research topic among researchers in virtual reality and computer animation . In this paper , an entire video clip is transformed into a small number of representative frames that are sufficient to represent a gesture sequence. These frames are the key frames that best represent the content of the sequence in an abstracted manner. In our method key frames are extracted using histograms [1] . Using these key frames only , entire gesture sequence has to be reconstructed and animated by extracting some features from each key frame . Separation of the hand from the arm is done using __ Distance Transform ‘‘ followed by largest circle fit method . After separating hand from the arm , metatarsophalangeal (MP) joints of the hand are extracted using the corner detection technique .

Finger tips are found by using contour detection . By using extracted MP joint locations and finger tips as parameters , hand pose is estimated . The hand pose is estimated using the hand model with 27 degrees of freedom and by imposing some constraints of the hand . Thus , the 3 D hand model for a particular gesture in a key frame is obtained . Since , the estimation of hand pose is only done for key frames , we do not have any information about the pose of the hand for the frames between two consecutive key frames.

For this, the hand model for the frames in between the key frames is obtained by interpolation. Finally , the gesture sequence is animated using these models and the extracted hand parameters.

The organization of the rest of the paper is as follows .In sect . 2 we present our hand gesture animation system . Sect. 2.6 shows the approach of key frame extraction . Finally, Experimental results are shown in sect. 3 and our conclusion in sect. 4.

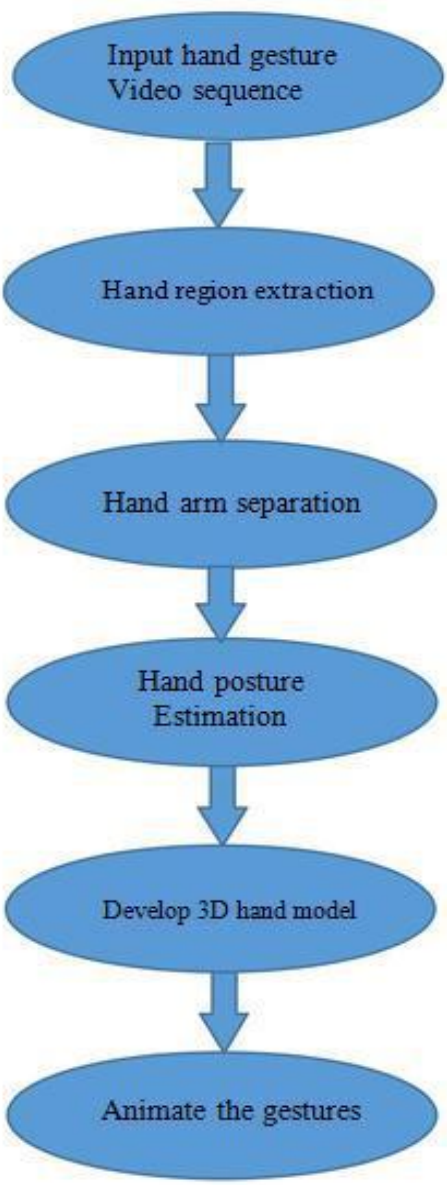

Figure 1 Systematic approach of the project

2 HAND GESTURE ANIMATION SYSTEM

The above systematic block model gives the overview of hand gesture animation system. From the input gesture video sequence, hand region is extracted by applying skin color-based segmentation. Next, hand is separated from the arm. After extracting the key frames, hand parameters are extracted, which are subsequently used for construing the hand pose. Finally, the animation is done by exporting all the hand parameters to the animation framework which was developed in the OpenGL platform.

2.1 Human hand modeling

The human hand is an articulated structure with about 27 degrees of freedom and hand changes its shape in various ways by its joint movements [2, 3]. Hand images change by both finger movements and hand movement as a whole . However , it is also highly constrained i.e., as the hand is incapable of making arbitrary gestures . Analysis of hand constraints and inter -relationship is essential to avoid unrealistic motions during hand animation . So, it is important to implement the 3D model of human hand in accordance with its constraints in OpenGL . There are many examples of such constraint [4,5].

2.2 HUMAN HAND SKELETAL MODEL

The human hand is highly articulated. To model the articulation of fingers, the kinematical structure of hand should be modeled. Such a hand kinematical model is shown in Figure with the names of each joint. This kinematical model has 27 Degrees of Freedom (DoF). Each of the four fingers has four DoF. The distal interphalangeal (DIP) joint and proximal interphalangeal (PIP) joint each has one DoF and the metacarpophalangeal (MCP) joint has two DoF due to flexion and abduction. The thumb has a different structure from the other four fingers and has five degrees of freedom, one for the interphalangeal (IP) joint, and two for each of the thumb MCP joint and trapeziometacarpal (TM) joint both due to flexion and abduction. The fingers together have 21DoF. The remaining 6 degrees of freedom are from the rotational and translational motion of the palm with 3 DoF each. These 6 parameters are ignored since we will only focus on the estimation of the local finger motions rather than the global motion .Articulated local hand motion, i.e. finger motion, can be represented by a set of joint angles θ, or the hand state.

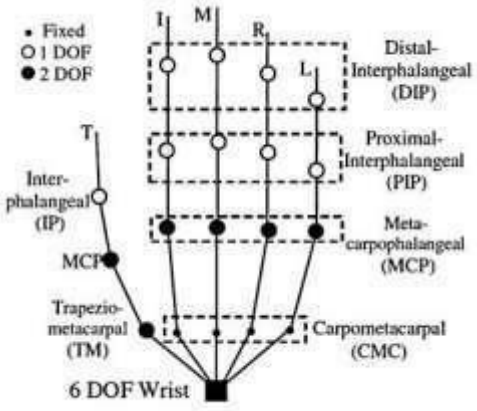

Figure 2 Skeletal Model of hand

[1] **2.3 Constraints overview**

Hand/finger motion is constrained so that hand cannot make arbitrary gestures. There are many examples of such constraint.

2.3.1 Type-I constraints

This type of constraint refers to the limits of the range of finger motions as a result of hand anatomy. We have only considered the range of motion of each finger that can be achieved without applying external forces such as bending of fingers backward using the other hand. This type of constraints are usually represented using the following inequalities:

$0° \leq \theta$ MCP- F$\leq 90°$

$0° \leq \theta$ PIP- F$\leq 110°$

$0° \leq \theta$ DIP $\leq 90°$ and

$-15° \leq \theta$ MCP-AA $\leq 15°$

A commonly adopted constraint states that middle finger displays little abduction/adduction motions and the following approximation is made for middle finger:

θ MCP-AA=0°

This will reduce one DoF from the 21 DoF model . Similarly , the TM joint of thumb also displays limited abduction motion and will be approximated by 0 as well.

θ TM-AA=0°

2.3.2 Type-II constraints

This type of constraint refers to the limits imposed on joints during finger motions . These constraints are often called dynamic constraints and can be subdivided into intra-finger constraints and inter-finger constraints . The intra -finger constraints are the constraints between joints of the same finger. A commonly used one based on hand anatomy states that in order to bend the DIP joints , the PIP joints must also be bend for the index , middle , ring and little fingers . The relations can be approximated as following:

Θ DIP = 2/3 x θ PIP

We are now able to reduce the model with 21 DoF to one that is approximated by 15 DoF.

2.4 Hand region extraction

This is the first step towards hand parameter extraction ; hand region should be separated out of background. Hand region can be separated out from the complex background effectively by using skin color-based segmentation [6]. To do this, we check if the code is able to detect a pre-supplied video . If it is unable to capture the video , a message is displayed as the supplied video is unable to get read and the execution breaks at that point . On the other context , if the video get captured , then further processes are performed on the video as a sequence of frames to obtain the hand region . To do this , HSV (Hue , Saturation and Value) color spaces is used for identifying skin region . The region can be controlled by the HSV value . On such a particular HSV value , we are able to obtain only the exact hand region from the video.

Thus, we are able to calculate the HSV value of the hand region for all the other frames needed required to extract the hand region . Binary images may contain numerous imperfections . In particular, the binary regions produced by simple thresholding , are distorted by noise and texture[7]. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. These techniques can be extended to greyscale images . Morphological operations apply a structuring element to an input image(frame), creating an output image of the same size. In a morphological operation , the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors . By choosing the size and shape of the neighborhood , you can construct a morphological operation that is sensitive to specific shapes in the input image . The most basic morphological operations are dilation and erosion . Dilation adds pixels to the boundaries of objects in an image , while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image . In the morphological dilation and erosion operations , the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image . The rule used to process the pixels defines the operation as a dilation or an erosion . In this way the hand region is extracted without any imperfections . The output of the above operation is shown below:

2.5 HAND ARM SEPARATION

The arm has to be separated from the segmented image. ‗‗Largest Circle Fit‘‘ method is used for separating the arm [8], [9]. In this method, the largest circle which can be inscribed completely in the extracted hand region is found out. Ultimately palm region is the widest part of the hand. So, the largest circle will definitely be lying inside the palm. Subsequently, a tangent to that circle is drawn and the portion of the hand right to that tangent is found out. The radius and center of the largest circle can also be determined by using ‗‗Distance Transform‘‘. The distance transform is

an operator normally only applied to binary images. The result of the transform is a graylevel image that looks similar to the input image, except that the gray level intensities of points inside foreground regions are changed to show the distance to the closest boundary from each point[10]. Distance Transform ‘‘ gives the skeleton of the input object . An efficient implementation of distance transform can be done using Chamfer mask . This algorithm uses small masks containing integer approximations of distances in a small neighborhood thereby converting of distances in a small neighborhood thereby converting to an approximate of distances in a small neighborhood thereby converting is called distance transformation (DT).

Figure 3 Extracted hand region using skin segmentation from the video input.

Figure 4 Distance transform of edge image

2.5.1 Following steps are done to find the largest circle.

• Find the edge of the input hand image using

Canny edge detector. (The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images)

• Find the distance transform of the edge image.

• Normalize the distance transformed image into the range [0, 255].

• Search for the pixel having an intensity value 255.

This point is the center of the largest circle to be inscribed in the hand.

• Find the edge pixel of the hand, which is at a smaller distance from the estimated center of the circle.

• Radius of the largest circle is the distance of the point obtained in the previous step to the center of the circle. Because of normalization and geometry of hand, there may be more than one pixel having an intensity of 255. To solve this ambiguity, the center of mass of all the pixels having an intensity of 255 may be considered. This centroid may be considered as the center of largest circle inscribed in the palm region. Then, the pixels around the center of the palm will have the intensity value 255 after normalization. amount of memory needed for video data processing and complexity greatly. This implies that there is very small change in the hand pose from frame to frame resulting in large amount of temporal redundancy. Hence, an entire video sequence/clip can be compressed by a large amount by discarding frames which do not show much change in the hand pose and keeping only those frames in which the change in the hand pose is quite significant/ notable. These frames are referred to as key frames. Hence, the entire gesture sequence can be summarized by using the key frames. Subsequently, information about hand shape parameters are extracted from these key frames only. Overall, this greatly reduces the memory requirement and the total training time for storing gesture description of every sign.

Figure 5 Edge of the hand image using canny edge detection method

2.6 Key frame extraction using color histogram method

Key-frame extraction from video data is an active research problem in video object recognition and information retrieval. Key-frame refers to the image frame in the video sequence which is representative and able to reflect the summary of a video content. By using the key-frame it is able to express the main content of video data clearly and reduce the

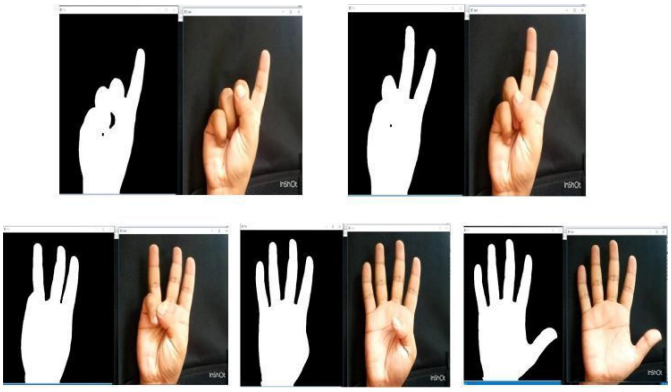

Figure 6 Key frames obtained from the input video

2.7 Hand parameter calculations

2.7.1 MP joint extraction by Harris corner detection method

The hand parameters include information about different angles of different fingers, thus describing the hand pose and shape in each key frame . Subsequently the pose of the hand can be estimated from these parameters . To do this , the first step is the identification of the MP joints of the hand. This is done with the help of corner detection technique. The procedures begin by finding the distance transform of the palm region and then obtain an image by thresh holding where some sharp points correspond to the MP joints of the fingers. The Harris corner detector is a popular interest point detector due to its strong invariance to: rotation, scale, illumination variation and image noise. The Harris corner detector is based on the local auto-correlation function of a signal; where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions. Since, in order a point to be a corner point, it should be an edge image. We can use Canny edge detection technique to obtain edge of the hand. Now, in order to know whether or not a particular point on the edge is a corner point, observe how the variation is present at that point in all directions.

2.7.2 K-curvature

After detecting all the corner points that includes the MP joints in the input threshold image this curvature method gives all such points where there is a change in the contour direction. Hence, there may be some extra points in addition to MP joints. To eliminate unnecessary points, we have used a method based on ‗k-curvature''. As MP joints correspond to a very sharp point, the curvature value is very small at the MP joints. Whereas, the curvature is very high at other corners points. So, we can eliminate false corners by thresholding the curvature value. The technique used to eliminate unnecessary corners points to detect valid MP joints.

2.7.3 Finger tips detection

As the MP joints are located, now finger tips can be found easily. To find the finger tips, the fingers from the palm are separated out [11]. Next, the contours of the separated fingers are found out. For each contour, centroid (center of mass) is estimated and the nearest MP joint from the corresponding centroid for all the contours is found out. Finally, we can find the point on the contour, which is farthest from the MP joint. All these points are the tips of the fingers. Obviously, the tip obtained from a contour and the nearest MP joint belong to a particular finger. So, the distance between these two points gives the finger pose length.

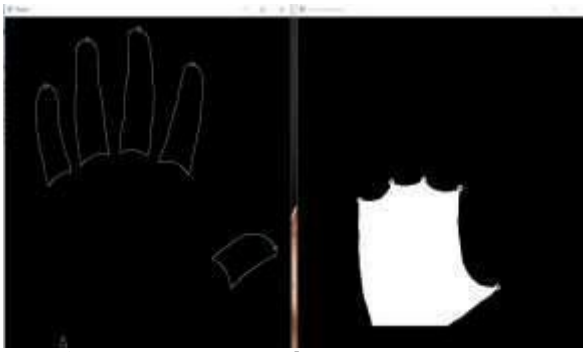

Figure 7 Finger tip and MP joints extractions

2.8 Hand posture estimation

The information of MP joints and finger tips can now be utilized to determine finger joint angles. The joint angles of the fingers are calculated as the distance between MP joint and the corresponding fingertip. For this, the hand modelling constraints is used. Let, ‗L'' be the original length of the finger. Similarly, ‗h1'' is the joint angle corresponds to flexion of the MP joint, ‗h2'' is the joint angle corresponds to PIP joint and ‗h3'' is the angle corresponds to DIP joint of finger. And let ‗Lobs'' be the observed length, i.e., distance between MP joint and the corresponding fingertip. In this, the known parameters are length of the finger (L) and observed Length (Lobs). Apparently, M, P, D and T represents MP joint, PIP joint, DIP joint and Tip of the finger. It is assumed that the lengths of all the joints of a finger are of equal length and MP, PIP, DIP and fingertip (T) lie in the same plane. The finger plane is aligned with x–y plane, such that MP joint coincides with the origin of the reference coordinate system. Here, finger plane is the plane passing through M, P, D and T. Assume that, each phalangeal is of length equal to one third of the total length of the finger[5].

$Tx = L/3(\cos(\theta1) + \cos(\theta1 + \theta2) + \cos(\theta1 + \theta2 + \theta3))$

$Ty = L/3(\sin(\theta1) + \sin(\theta1 + \theta2) + \sin(\theta1 + \theta2 + \theta3))$

Therefore, we can express Lobs as,

$Lobs = L/3(\cos(\theta1) + \cos(\theta1 + \theta2) + \cos(\theta1 + \theta2 + \theta3))$

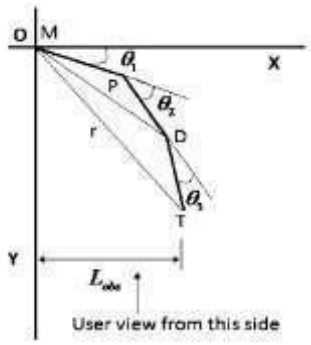

Figure 8 MP joints, PIP joint, DIP joint and Tip of a finger.

Let us consider the triangle, MPD. By using ‗Cosine' rule and ‗Sine' rule, the following relations can be obtained.

$(MD)\ ^2 = (L/3)^2 + (L/3)^2 – 2(L/3)(L/3)\ \text{Cos}\ (\Pi-\theta2)$

$\frac{\text{Cos}\ (\angle MDP) = ((L/3)^2 + (MD)^2 - (L/3)^2)/2(L/3)(MD)}{\text{Sin}\ MDP} = \frac{}{\text{Sin}\ (\Pi-\theta2)}$

(L/3) (MD)

Again, let us consider the triangle, MDT. By using ‗Cosine' rule for this triangle we can obtain the following relations. $\text{Cos}\ (\angle MDP) = \text{Cos}\ (\Pi-\theta2-\angle MDP)$

$r^2 = (MD)^2 + (L/3)^2 - 2\ (MD)\ (L/3)\ \text{Cos}\ (\angle MDP)$ We can also apply some of the constraints, given in Type-II constraints of hand model. So, θ3 is replaced by 2/3(θ2). So, θ3 can be removed from the above relations. Subsequently, we can solve for θ2. After solving for θ2, we use the equation for Lobs to solve for θ1. For each finger, by proceeding in the similar way, we can find joint angles of all fingers.

2.9 Hand gesture animation

After estimating the flexion angles of MP, PIP and DIP joints of all fingers, we can export these angles as parameters to 3D hand model developed in OpenGL. We can perform the animation of gestures using these parameters. Since, the extraction of the parameters has been done on the key frames only; we do not have any information about the pose of the hand, in the frames between the key frames. To give an illusion of continuous animation by presenting the different positions of hand, we should be able to estimate the pose of hand in these intermediate frames. This is called interpolation. As the name suggests, key frame interpolation attempts to generate new frames by interpolating values between two existing frames. One of the simplest method is linear interpolation explained below. For every single vertex that is in starting frame, we also have one in the ending frame and thus we are able to do linear interpolation between two frames. This doesn't apply only for vertex positions - this way, we can also animate normal or even texture coordinates. However, having only two key frames the animation would really not be good enough . Imagine an animation where you rotate something . For example , if we could rotate our heads by 180 degrees , the animation of two key frames would look like this: The face would get sucked into the head itself and then reappeared on the other side of head . So what we learned from here is that when making key frame animation , two key frames are just not enough . But if we approximate animation by 20 frames or so and then we would interpolate between keyframes consecutively one by one, the animation will look smoother and better.

RESULT

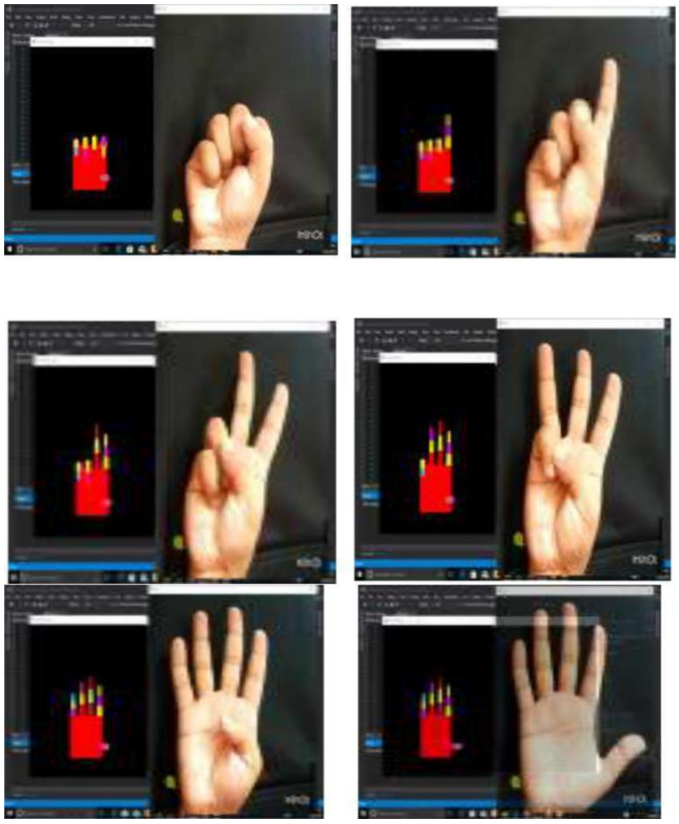

Figure 9 Final Reconstruction of the original gestures on the right and the reconstructed gestures on the left.

3 EXPERIMENTAL RESULTS

The gesture animation module was implemented using the image processing library of OpenCV 2.1 in Microsoft Visual Studio 2015 version 14 .0 IDE in a window - based system .Whereas , gesture animation part was implemented in the OpenGL platform CodeBlocks version 16 .01 . The hand model was developed using cylinders and cubic in OpenGL . A Web camera is used for capturing the hand movements . The results are satisfactorily obtained in the above mentioned setup .The results of the skin color segmentation are shown in Fig. 3, whereas Fig . 6 gives the key frames obtained from the gesture video sequence.

The results of MP joints and fingertip detection using corner detection technique are shown in Fig. 7. MP joints are obtained by using k-curvature method . Finally , Fig . 9 shows the animation of single-handed gestures.

4 CONCLUSION

Previous research works are included specifically in research papers [1] and [12], which extensively deals with this very specific research topic . For our system we research various approaches and its methods, used for hand gesture recognition.

Some assumptions are made in the very beginning and we stuck with them as they made the project more achievable .In terms of image processing , the various basic techniques for extracting important information from image and techniques like adaptive thresholding , calculating contours are studied and various colour spaces and their suitability for various applications has been explored. In order to implement these techniques and also the various functions provided by OpenCV has been explored as well . In this report , we analyze the hand gestures having only local motions i. e., palm is kept fixed and gestures are created using finger movements only. However , the proposed technique can be extended with some modifications to include more natural form of gestures i.e., global motions of palm and arm . For this , along with the MP joints and fingertips of fingers , wrist and elbow position of the hand should also be considered . Future works also include the segmentation of the hand in a more cluttered background and tackling the problem of interference among the fingers for gesture animation.

REFERENCES

[1]. M . K . Bhuyan , V . Venkata Ramaraju , Yuji Iwahori Hand gesture recognition and animation for local hand motions 21st March, 2013.

[2]. Lin J, Wu Y , Huang TS , ― Modeling the constraints of human hand motion‖, In: Proceedings of workshop on human motion, pp 121–126, 2000.

[3]. Guan H, Chua CS, Ho YK (2001) 3D hand pose retrieval from a single 2D image . Proc Int Conf Image Process 1:157–160

[4]. Wu Y, Huang TS, ― Hand modelling, analysis, and recognition for visionbased human computer interaction ‖, IEEE Signal Process Mag 18:51–60, 2001.

[5]. Lee J, Kunii TL (1995) Model - based analysis of hand posture. IEEE Comput Graph Appl 15(5):77–

[6]. Tan W, Wu C, Zhao C, Chen S (2009) Hand extraction using geometric moments based on active skin color model. In: Proceedings of IEEE international conference on intell computingand intelligent systems, pp 468–471

[7]. Teng X, Wu B, Yu W, Liu C (2006) A hand gesture recognition system based on local linear embedding. In: Proceedings of IEEE conference on robotics, automation and mechatronics,vol 16 , pp 1–6

[8]. Dong G, Yonghua Y, Ming X (2002) Vision - based hand gesture recognition for human -vehicle interaction . In: Proceedings of 7th international conference on control, automation , robotics and vision, pp 1–4.

[9]. Amayeh G, Bebis G, Erol A, Nicolescu M (2007) A new approach to hand -based authentication . In: Proceedings of biometric technology for human identification

[10]. Borgefors G (1986) Distance transformations in digital images . Comput Vision Graph Image Process 34:344–371

[11]. Schwarz C, da Vitoria Lobo N (2005) Segment-based hand pose estimation. In: Proceedings of the 2nd Canadian conference on computer and robot vision, vol. 20 pp 42-49

[12]. Verma V , Ghosh D (2005) Hand gesture reconstruction and animation . In: Proceedings of 2 nd international conference on artificial intelligence (IICAI-05), pp 537–555conference on artificial intelligence (IICAI-05), pp 537–555

