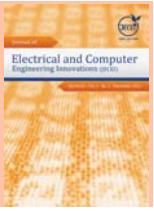
SRTTU

Journal of Electrical and Computer Engineering Innovations

JECEI, Vol. 1, No. 1, 2013

Regular Paper



3D Hand Motion Evaluation Using HMM

A. Safaei[1,*] and M. Jahed[2]

[1]School of Science Engineering, Sharif University of Technology, International Campus, Kish Island, Iran

[2]School of Electrical Engineering, Sharif University of Technology, Tehran, Iran

[*]Corresponding Author: amin_safaei@alum.sharif.edu

ARTICLE INFO

ARTICLE HISTORY:
Received 5 June 2013
Revised 24 June 2013
Accepted 8 July 2013

KEYWORDS:
Machine Vision
Stereo Vision
Motion Recognition
Video Processing
Image Processing
HMM (Hidden Markov Model)

ABSTRACT

Gesture and motion recognition are needed for a variety of applications. The use of human hand motions as a natural interface tool has motivated researchers to conduct research in the modeling, analysis and recognition of various hand movements. In particular, human-computer intelligent interaction has been a focus of research in vision-based gesture recognition. In this work, we introduce a 3-D hand model recognition method that offers flexible and elaborate representation of hand motion. We used landmark points on the tips and joints of the fingers and calculated the 3-D coordinates of these points through a stereo vision system followed by a Hidden Markov Model (HMM) to recognize hand motions. Experimentally, in an effort to evaluate the formation of hand gestures similar to those used in rehabilitation sessions, we studied three evolving motions. Given the natural hand features and uncontrolled environment, we were able to classify and differentiate unnatural slowness or rapidness in the performance of such motions, ranging from 45% to 93%.

1. INTRODUCTION

Recently, recognition of hand gestures and hand motion tracking has become an important issue in the field of human-computer interaction. Many researchers have tried or are trying to create a method to capture human gestures and hand motions using a camera [1]. The main problem in gesture recognition is to establish a fully automated and adaptable hand gesture recognition system. In general, such studies are divided into two groups: "Data-Glove based" and "Vision based."

The Data-Glove based method uses sensors to convert a hand motion to digital multi-parametric data. This method is rather expensive and elaborate. In contrast, the Vision based methods require only a camera [2], thus realizing a more natural human-computer interaction in this regard. These systems tend to complement biological vision by describing artificial vision systems that are implemented in software and/or hardware. This design poses a challenging problem as these systems need to be background invariant, environment-lighting insensitive, and person- and camera-independent to achieve real-time performance. Moreover, such systems must be optimized to meet the requirements, including accuracy and robustness [2, 3].

In Vision based methods, the motion of the hand is recorded by the camera. This input video is decomposed into a set of features taking individual frames into account [2]. To separate the hand from other body parts or background objects, it is necessary to use some kind of filtering. The Vision based methods are divided into two categories: "3-D Hand Models" and "Appearance Hand Models".

The 3D hand models rely on the 3D kinematic description of the hand, providing an elaborate depiction of the hand. To capture and generate a 3D hand model, such studies generally utilize stereo cameras and IR sensors. However, the appearance hand model approach simply utilizes image features to model the visual manifestation of the hand. Vision-

based approaches are gaining more interest with the advantages of being intuitive, device-independent and non-contact based [4-6].

A common approach to describe human motion is to use a state-based model, such as a Hidden Markov Model (HMM) [7].

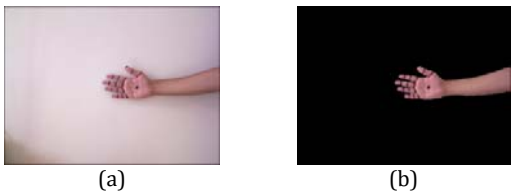
As a special case of human-computer interaction and rehabilitation, several constraints are imposed. These include complexities, background, variable lighting conditions, transforming gesture structures, real time implementation and the dependency on user and device characteristics. Numerous techniques on gesture-based interaction have been proposed, but almost no published works fulfill all the requirements stated above.

Our methodology began with a basic framework that generated a 3D model of the objects; we used stereo vision and 3-D reconstruction techniques to recover the 3D hand posture. A segmentation approach was used to isolate the object of interest from other objects and to extract features of the objects. A classifier was then used and was trained based on the feature vectors of the object for a number of classes and could then be used to classify new unseen objects. Finally, the Hidden Markov Model (HMM) was used to recognize motions.

The remainder of this paper is organized as follows. First, the segmentation process is discussed. Then, the theory behind stereo vision and the implementation steps as well as our basic framework for feature extraction is described. Next, the classification and recognition methods are explained. In the experimental results section, a set of classification and recognition experiments is described and analyzed. Finally, a conclusion and suggestions for future work are provided.

2. METHOD

Our methodology was developed for single 3D object recognition, as required for rehabilitation applications. It cannot accommodate objects in occlusion. Our methodology consists of segmentation, construction of the 3D object, classification and recognition. In this section, the theory and implementation of the proposed method are described.



(a) (b)

Figure 1: (a) Original hand image and (b) segmented hand

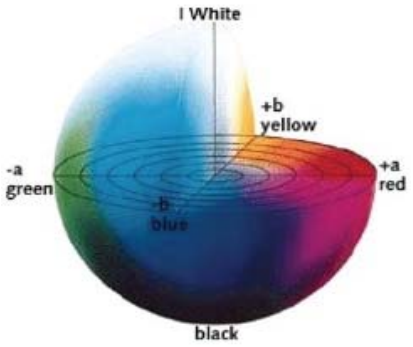


Figure 2: L*a*b* color space [11]

A. Segmentation

Segmentation is a process that isolates the hand from the background. It is the most important step in every hand gesture recognition system. All of the following steps depend entirely on this step.

Clustering is a method to separate groups of objects from others. K-means clustering is a method of cluster analysis that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean [8]. It is essential to define the number of clusters that are needed and the distance metric. In this part, we used the K-means method to segment skin color. During the implementation process, we transferred the image to the L*a*b* color space (CIELab); with this technique, we can visually distinguish colors from one another. The L*a*b* color space consists of 'L*,' which represents the lightness coordinate, 'a*,' which represents the red/green coordinate and 'b*,' which represents the yellow/blue coordinate. We use the color information in the 'a*' and 'b*' layers and the Euclidean distance metric to determine the difference between two colors. Figure 1 shows the result of hand segmentation and figure 2 represents the L*a*b* color space model [9-11].

B. 3D Hand posture recognition

Binocular vision is defined as vision from two eyes where the data being perceived from each eye is partially overlapped. The overlap from the two different views is used in biological vision to perceive depth [12]. Stereo vision uses binocular vision to generate the 3D model of an object. We used this method to find and generate a disparity map, which consists of the difference in the location of an object that is viewed from two different cameras. A simple stereo vision system consists of two or more cameras that are horizontally aligned and separated by a distance called the baseline.

Based on the images that are captured from the left and right cameras, we constructed a 3D model. During the implementation process, solutions to several problems were proposed using the following four

steps:

1. Calibration: Remove radial and tangential lens distortion.
2. Rectification: Adjust the angles and distances between the cameras.
3. Disparity Map: Find the same features in the left and right camera views (a process known as correspondence). The output of this step is a disparity map [13].
4. Projection: Turn the disparity map into distances by triangulation.

C. Camera model

The simple model of the camera consists of these components: an image plane R, the optical center C and the focal length [14]. Referring to Figure 3, the line perpendicular to R that goes through the optical center C is called the optical axis [14]. The point that intersects the plane R is called the principal point. There are two coordinate frames of interest in the pinhole camera model: the orthonormal 2D plane in the image plane and the camera coordinate system, which is centered at the optical center with the plane defined by two of its axes being parallel to the image plane [14]. Figure 4 shows the pinhole camera model with the image plane in front of the optical center. Equations (3) and (4) show the relationship between point P in the camera coordinate frame and its projection P' in the image.

$$P = (X, Y, Z)^T \tag{1}$$

$$P' = (x_c, y_c, f) \tag{2}$$

$$x_c = f\frac{X}{Z} \tag{3}$$

$$y_c = f\frac{Y}{Z} \tag{4}$$

$$\begin{pmatrix} Zx_c \\ Zy_c \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = p \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{5}$$

This relationship can be written with homogeneous coordinates as a matrix multiplication operation where P is the projection matrix. If P' is normalized by dividing by the third coordinate the point, it is known as the normalized image plane [13, 14].

D. Intrinsic camera parameters

The perspective projection can be defined using equation (6), where Q represents a 3-D point, q describes a 2-D image point and M is defined as the intrinsic camera matrix. Parameters f and C in the equation (7) represent the focal length and the principal point, respectively [12, 13].

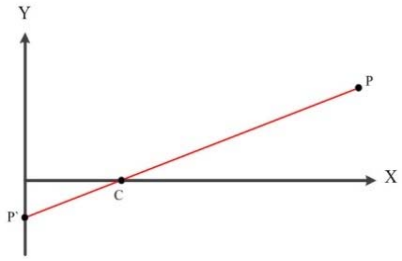


Figure 3: Simple camera model

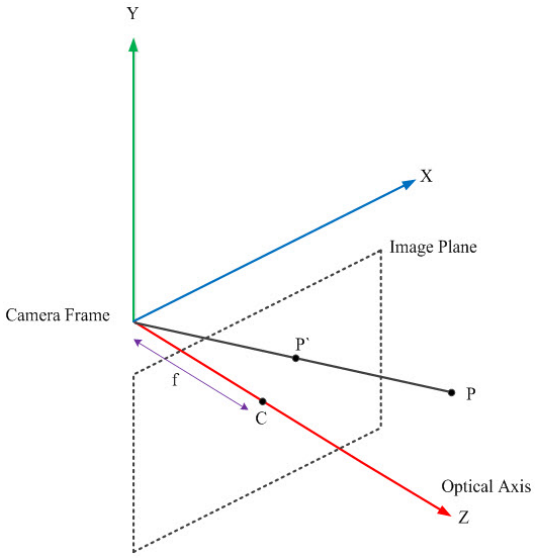


Figure 4: Pinhole camera with image plane

$$q = MQ \tag{6}$$

$$q = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \tag{7}$$

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{9}$$

E. Lens distortion

In practice, we cannot find and use a lens with no distortion. The distortion of the lens is mainly due to intrinsic manufacturing processes. There are two models for lens distortion. The first one is for Radial distortion, which arises as a result of the shape of the lens and the second one is a tangential distortion, which arises from the assembly process of the camera. We can model the radial distortion with equations (10) and (11), where r is the distance from the point in the image plane to the optical center and < k1, k2, k3 ... >

are the coefficients of the polynomial series. Figure 5 shows the perspective model with the radial distortion [13, 14].

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (10)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \quad (11)$$

We can also model the tangential distortion with equations (12) and (13), where p1 and p2 are the coefficients for the tangential distortion.

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)] \quad (12)$$

$$y_{corrected} = y + [2p_2 x + p_1(r^2 + 2y^2)] \quad (13)$$

To find the distorted point in the image plane, we add both radial and tangential distortions to the projection.

$$x_d = x_c + x_{corrected-Radial} + x_{corrected-\tan gential} \quad (14)$$

$$y_d = y_c + y_{corrected-Radial} + y_{corrected-\tan gential} \quad (15)$$

F. Extrinsic camera parameters

Extrinsic camera parameters consist of the translation vector T and the rotation matrix R. If the point of interest is located in the coordinated frame of the camera, equation (5) can be used; however, if the point is in another coordinate frame, it needs to be transferred to the camera coordinate frame. A point in the reference coordinates frame can be transferred to the camera frame using equation (16). The relationship between the camera coordinate frame and another one is defined as the extrinsic camera parameters [13, 15, 16].

$$p_c = R^T(p_w - T) \quad (16)$$

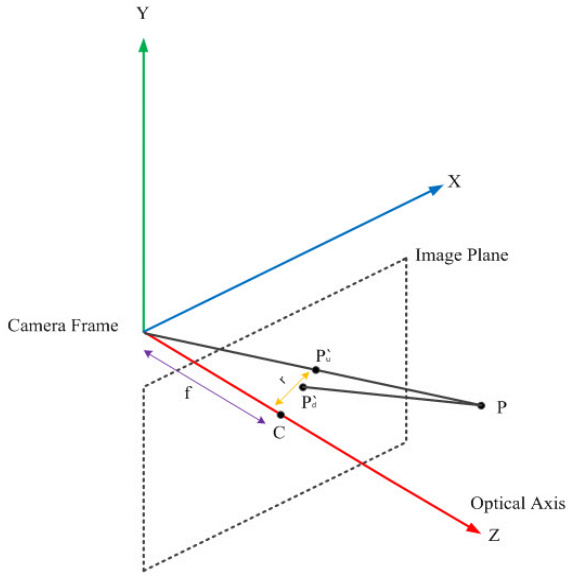


Figure 5: Perspective model with radial distortion

G. Epipolar geometry

Epipolar geometry is the geometry of the stereo vision [17]. When two cameras are placed in a distinct position, they share a common viewing volume. Figure 6 shows this epipolar geometry.

A point X can be visible in both the right and left cameras, where we consider it in the left camera with X_L and in the right camera with X_R. The image points X_L, X_R and X are coplanar with center points of the cameras, and this plane is called the epipolar plane. The line between the focal point of the left and right camera is called the epipolar line, and the intersections of this line with the image planes eL and eR are called the epipoles. The important feature of epipolar geometry is that if we consider any point in the left or right image plane, we have an epipolar line in the opposite plane and we can use this rule to estimate a depth map. When considering point X_L in the left plane, we cannot determine which of the points X_1,X_2 or X_3 belong to the point X_L. When we know the correct correspondence to X_L in the right image plane, we can then find the depth point X.

H. Essential and fundamental matrices

When we want to map a point in one image and epipolar lines in the other we use an Essential Matrix, E and a Fundamental Matrix, F. Consider that we want to map a point in the left image to an epipolar line in the right image. This process consists of two steps. In the first step, the point in the left image is mapped to some point in the other image on the epipolar line and in the next step, the epipolar line is obtained while we join the right point to the epipole. The difference between E and F is that E expresses the relation between the left and right camera in the 3-D physical space while F expresses the relation in pixel coordinates. We can use equations (17) and (18) to find matrices F and E based on the locations of point P in the left and right cameras [12-14].

$$p_r^T E p_l = 0 \quad (17)$$

$$q_r^T F q_l = 0 \quad (18)$$

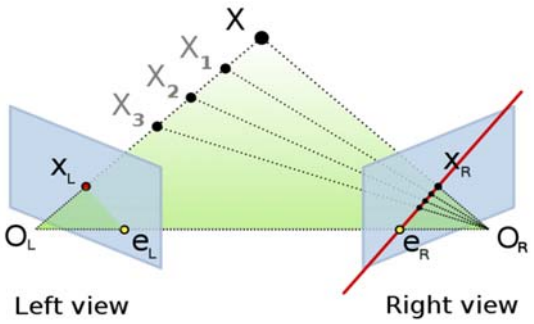


Figure 6: Epipolar geometry [17]

I. Calibration

Calibration is the process used to determine the intrinsic and extrinsic parameters of the camera. In the previous section, we described how to compute E and F. Equations (19) and (20) express how we can find the point P, on the right and left camera. We can also use equation (21) to express the relationship between points PL and PR. Using these three equations, we can arrive at equation (22) for rotation and equation (23) for translation [13, 18].

$$P_l = R_l P + T_l \tag{19}$$

$$P_r = R_r P + T_r \tag{20}$$

$$P_l = R_T (P_r - T) \tag{21}$$

$$R = R_r (R_l)^T \tag{22}$$

$$T = T_r R T_l \tag{23}$$

In practice, we use OpenCV built in calibration function and a chessboard. Figure 7 illustrates one frame of the calibration process.

J. Corresponding point and 3D construct

To construct a 3-D model of the hand, we used a re-projection matrix Q (24). After finding 2D points of interest and their associated disparity d, we can use equation (25) to project the point into 3D and then use X/W, Y/W and Z/W to determine the 3-D coordinates.

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c_x')/T_x \end{bmatrix} \tag{24}$$

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \tag{25}$$

The parameters of equation (24) relate to the left image except parameter cx ′, which relates to the right image. If the principal rays intersect at infinity, then cx = cx ′ and the term in the lower right corner is 0 [13].

K. Feature vector

The human hand is an articulated object it can be described in this way: the base is a palm and five fingers are attached to the palm. Basically, each finger has four DOFs, two for the metacarpophalangeal (MP) joint and abduction (ABD) and another two for the proximal interphalangeal (PIP) and distal interphalangeal (DIP) joints. Figure 8 shows the hand joints [19, 20].

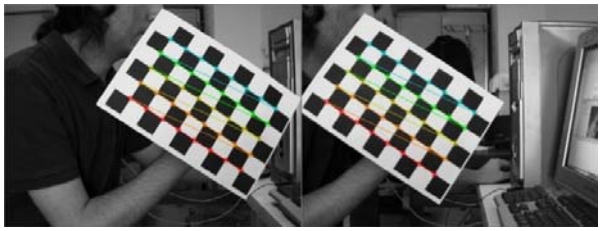


Figure 7: Calibration procedure

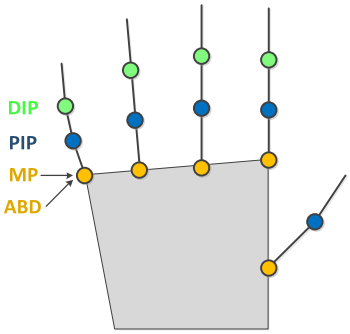


Figure 8: Hand joints

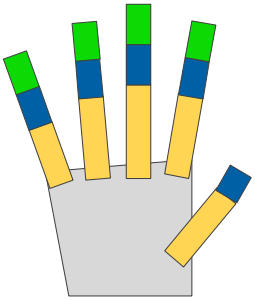


Figure 9: Cardboard model

When viewed from the direction orthogonal to the palm, the hand could be modeled by a cardboard model, in which each finger could be represented by a set of three connected planar patches. The length and width of each patch should be adapted to individual people. Although it is a simplification of a real hand, it offers a good approximation for motion captured under this specific viewing direction [19, 20]. The cardboard model is shown in Figure 9.

If we want to determine the hand position, we only need to know the position of the tip of the fingers and the finger joints. In this work, we placed markers on these landmark points and generated the 3D position of these points using the stereo vision method. When the 3D position of each joint was determined, we added two parameters based on the 3D position: the Azimuth Angle, θ, which is defined as the angle between the positive xz plane and the projection of n onto the x plane and the Elevation Angle, φ, of n, which is defined as the angle between the x plane and the vector n (Figure 10).

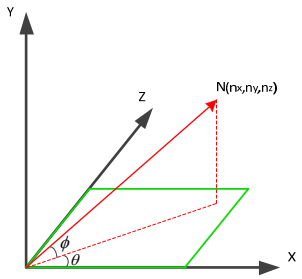


Figure 10: Azimuth Angle θ and Elevation Angle φ

$$\theta = \arctan(\frac{n_z}{n_x}) \quad (26)$$

$$\phi = \arctan\left(\frac{n_y}{\sqrt{n_x^2 + n_z^2}}\right) \quad (27)$$

where θ = [-π,π] and ϕ = [-π/2,π/2]. The feature for each point consists of 5 parameters and we have 95 parameters for each frame or hand pose.

$$P = \{P_x, P_y, P_z, P_\theta, P\phi\} \quad (28)$$

L. Classification

To classify each hand pose based on the feature vectors defined in the previous section, we used a Nearest Neighbor method. The Nearest Neighbor method originated in the statistics field and was first considered for rule production by Fix and Hodges (1951), who performed an initial analysis of the properties of k-nearest neighbor systems. This method measures the distance between the new data and the trained data. The new data are then classified based on its nearest neighbor. For numeric attribution, this algorithm uses the Euclidean Distance method, where each data point is considered as a point in an n-dimensional feature space. This algorithm considers that for a given data point in the feature space, the surrounding area will share the same class. The Euclidean function further assumes that all features are equally important and thus share the same scale in the feature space and that this scale is linear along each axis [21]. If the points P and Q are in n-space, the distance from P to Q is given by equations (29) and (30).

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \ldots + (q_n - p_n)^2} \quad (29)$$

$$d(p.q) = d(q,p) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad (30)$$

M. Hidden Markov Model

The HMM model is a statistical model that is successfully used in speech and gesture recognition [14]. It is composed of N state and M observation symbols where A is the probability of transitioning from state Si to Sj in one time-step, B is the observation symbol probability distribution and π is the initial state distribution.

$$\lambda = (A, B, \pi) \quad (31)$$

We define the specific HMM model for hand motion recognition by considering that hand motion consists of discrete hand poses and that each hand pose can be represented as a state. We define a bounded Left-Right model; in this model, each state can transmit to the next state or itself. Figure 11 represents the Left-Right model [22-24].

3. Experimental Results

The proposed hand gesture recognition system was tested to demonstrate its feasibility. The platform used was Microsoft Windows 7 on a PC with Intel Core 2 Duo Processor 2.66GHz and 4Gbytes main memory. Sony DFW-X71 and Logitech C300 webcams were used to capture images. The software development tools consisted of Microsoft Visual Studio 2010 with Computer Vision library OpenCV 2.2. We also used NNge implemented in weka for classification and the segmentation program was developed in Matlab R2010a.

Table 1 shows the results of recognition for three evolving motions. Given the natural hand features and uncontrolled environment, we were able to classify and differentiate unnatural slowness or rapidness in the performance of such motions. Figure 12 shows the proposed system and Figure 13 shows selected motions.

Table 1
Recognition Result

Motions	Motions			
	Slow at the beginning	Slow at the end	Fast at the beginning	Fast at the end
Motion 1	81.01%	93.47%	51.35%	62.79%
Motion 2	92.1%	91.36%	76.15%	82.32%
Motion 3	74.21%	70.21%	45.31%	51.86%

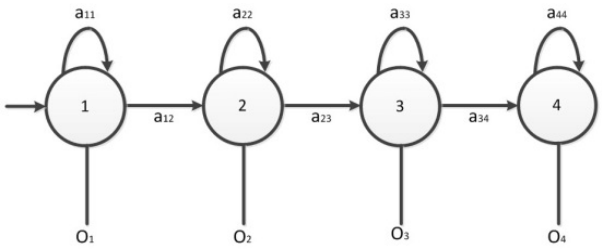


Figure 11: HMM Left-Right model

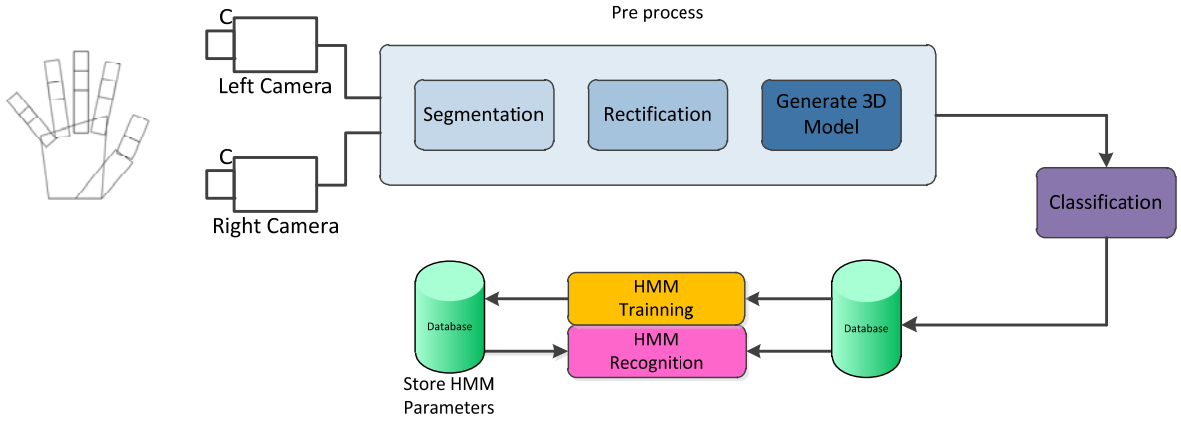


Figure 12: Proposed system

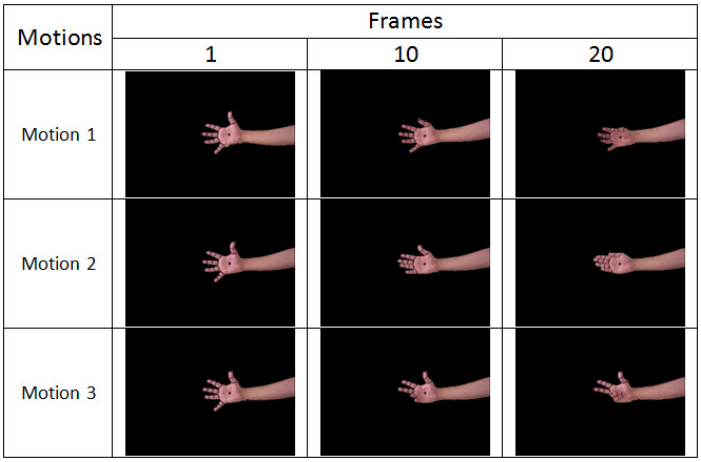


Figure 13: Motions

4. Conclusion

We proposed a framework and methodology for 3D object reconstruction and representation and discussed its application for 3D object recognition. The methodology was initiated by generating a 3D model and then extracting features. The classifier learned the characteristics of points of interest based on the extracted feature and classifies hand poses. Finally, HMM was used to evaluate and recognize the movements based on the rate of the evolving motions.

References

[1] Mehrdad Sangi, Mehran Jahed, "A Fast 3D Hand Model Reconstruction by Stereo Vision System," IEEE 2010.

[2] X. Zabulis, H. Baltzakis, A. Argyros, "Vision-based Hand Gesture Recognition for Human-Computer Interaction," Institute of Computer Science Foundation for Research and Technology - Hellas (FORTH) Heraklion, Crete, Greece.

[3] Pragati Garg, Naveen Aggarwal and Sanjeev Sofat, "Vision Based Hand Gesture Recognition," World Academy of Science, Engineering and Technology 49 2009.

[4] Ying Wu and Thomas S. Huang, "Hand Modeling, Analysis, and Recognition," IEEE Signal Processing Magazine,2001.

[5] Ali Erol, George Bebis, Mircea Nicolescu, "A Review on Vision-Based Full DOF Hand Motion Estimation," IEEE Human Motion Proceedings, 2000.

[6] Nilanjan Sarkar , "Human-Robot Interaction," Itech Education and Publishing, Vienna, Austria, 2007.

[7] Nilanjan Sarkar , "Human-Robot Interaction," Itech Education and Publishing, Vienna, Austria, 2007.

[8] Available in: http://en.wikipedia.org/wiki/Kmeans_clustering.

[9] Dana Elena Ilea and Paul F. Whelan, "Color Image Segmentation Using a Spatial K-Means Clustering Algorithm," IMVIP 2006 - 10th International Machine Vision and Image Processing Conference, 30 August - 1 September 2006, Dublin, Ireland.

[10] Tse-Wei Chen,Yi-Ling Chen, Shao-Yi Chien, "Fast image segmentation based on K-Means clustering with histograms in HSV color space," Multimedia Signal Processing, 2008 IEEE 10th Workshop on, 8-10 Oct. 2008.

[11] "Definition of CIELab Color Space," available in: http://www.optelvision.com/documents/optel-vision-s-explanation-on-cielab-color-space.pdf

[12] Nathaniel J. Short, " 3-D Point Cloud Generation from Rigid and Flexible Stereo Vision Systems," Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in Computer Engineering 2009.

[13] Gary Bradski and Adrian Kaehler, "Learning OpenCV," O'REILLY,2008.

[14] Ricardo A. Arango S, "Learning Action Primitives From 3D Stereo Vision Measurements," A thesis submitted to the University of Aalborg for the degree of Master of Engineering in Computer Vision and Graphics 2010.

[15] J. Lou, Q. Liu, T. Tan, and W. Hu, "Semantic interpretation of object activities in a surveillance system," In Proc. International Conference on Pattern Recognition, pages III: 777–780, 2002.

[16] N.M. Oliver, B. Rosario, and A.P. Pentland, "A Bayesian computer vision system for modeling human interactions," IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8):831–843, August 2000.

[17] Available in:

http://en.wikipedia.org/wiki/Epipolar_geometry.

[18] Roger Y. TSAI , "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off the-shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, VOL. RA-3, NO. 4, AUGUST 1987.

[19] Ying Wu, John Y.Lin, Thomas S.Huang, " Capturing Natural Hand Articulation," IEEE International Conference on Computer Vision, Canada, 2001.

[20] Makoto Kato, Yen-Wei Chen and Gang Xu, "Articulated Hand Tracking by PCA-ICA Approach," International Conference on Automatic Face and Gesture Recognition, IEEE 2006.

[21] Brent Martin, "INSTANCE-BASED LEARNING: Nearest Neighbor with Generalization," Department of Computer Science University of Waikato Hamilton, New Zealand, March 1995.

[22] Gernot A. Fink, "Markov Models for Pattern Recognition From Theory to Applications," Springer-Verlag Berlin Heidelberg 2008.

[23] Mahmoud Elmezain, Ayoub Al-Hamadi, J¨org Appenrodt, Bernd Michaelis, "A Hidden Markov Model-Based Continuous Gesture Recognition System for Hand Motion Trajectory," IEEE 2008.

[24] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, February, 1989.

BIOGRAPHIES

Amin Safaei was born in Tehran, on Apr 19, 1983. He got his BS at Islamic Azad University – Garmsar Branch and got his MSc from Sharif University of Technology – International Campus. He is working as a university lecture and researcher at the Islamic Azad University – Firouzkoh Branch. His research interests are Machine Vision, Video Processing, Digital Signal Processing and Reconfigurable System. His employment experience included; the HES Innovative Company; Tehran, Saba Niroo and EEI Institute. His special fields of interest are Implementing digital system and Machine Vision algorithms on a Reconfigurable System.
Email: amin_safaei@alum.sharif.edu

Mehran Jahed completed his EE Bachelor's degree in EE in 1982, at Purdue University, West Lafayette, IN and Masters and PhD in EE in 1987 and 1990, at the University of Kentucky. He was a post-doctoral fellow at the Center for Excellence for Biomedical Engineering at University of Kentucky from 1990 to 1993. Since 1993 he has been a faculty member of EE department at Sharif University of Technology. His research interests are in biomedical modeling and control, bio-robotics and prosthetic systems, virtual and augmented reality, and application of artificial intelligence in medicine and biology.
Email: jahed@sharif.edu