A Vision Based Dynamic Gesture Recognition of Indian Sign Language on Kinect based Depth Images

Geetha M*, Manjusha C†,Unnikrishnan P‡ and Harikrishnan R‡
geetha.m.amrita@gmail.com*,manjunharry@gmail.com†,unnikrishnanenator@gmail.com†
*Dept. of Computer Science and Engg.
Amrita School of Engineering
Amritapuri, Kollam, Kerala, India

***Abstract*—Indian Sign Language (ISL) is a visual-spatial language which provides linguistic information using hands, arms, facial expressions, and head/body postures. Our proposed work aims at recognizing 3D dynamic signs corresponding to ISL words. With the advent of 3D sensors like Microsoft Kinect Cameras, 3D geometric processing of images has received much attention in recent researches. We have captured 3D dynamic gestures of ISL words using Kinect camera and has proposed a novel method for feature extraction of dynamic gestures of ISL words. While languages like the American sign language(ASL) are of huge popularity in the field of research and development, Indian Sign Language on the other hand has been standardized recently and hence its (ISLs) recognition is less explored. The method extracts features from the signs and convert it to the intended textual form. The proposed method integrates both local as well as global information of the dynamic sign. A new trajectory based feature extraction method using the concept of Axis of Least Inertia (ALI) is proposed for global feature extraction. An eigen distance based method using the seven 3D key points- (five corresponding to each finger tips, one corresponding to centre of the palm and another corresponding to lower part of palm), extracted using Kinect is proposed for local feature extraction. Integrating 3D local feature has improved the performance of the system as shown in the result. Apart from serving as an aid to the disabled people, other applications of the system also include serving as a sign language tutor, interpreter and also be of use in electronic systems that take gesture input from the users.**

***Keywords*—Dynamic Gestures, Indian sign Language, Microsoft Kinect, Axis of Least Inertia, Principal Component Analysis**

I. Introduction

Gesture recognition is an area of active research in computer vision. While speech recognition has made rapid advances, sign language recognition is lagging behind. Communication with or by the deaf and dumb are based on the sign language followed by a country. Not everyone is well versed with the sign language followed in their country. As a result, the deaf and the dumb find it difficult to communicate with the people around them. The motive of our project is to make communication easy for the deaf and dumb thus taking this project to the level of serving the society. The principal constituent of any sign language recognition system is hand gestures and shapes normally used by deaf people to communicate among themselves. A gesture is defined as a energetic movement of hands and creating signs with them such as alphabets, numbers, words and sentences. Gestures are classified into two type static gesture and dynamic gestures. Static gesture refer to certain pattern of hand and finger orientation where as dynamic gestures involve different movement and orientation of hands and face expressions largely used to recognize continuous stream of sentences. Our method of gesture recognition is a vision based technique which does not use motion sensor gloves or colored gloves for the system to recognize hand shapes. A complete gesture recognition system requires understanding of hand shapes, finger orientations, hand tracking and face expressions tracking.

Accordingly sign language recognition systems are classified into two broad categories: sensor glove based and vision based systems.

The first category requires signers to wear a sensor glove or a colored glove. The wearing of the glove simplifies the task of segmentation during processing. Glove based methods suffer from drawbacks such as the signer has to wear the sensor hardware along with the glove during the operation of the system.

In comparison, vision based systems use image processing algorithms to detect and track hand signs as well as facial expressions of the signer, which is easier to the signer without wearing gloves. However, there are accuracy problems related to image processing algorithms which are a dynamic research area.

For Recognition, methods that take into account both the global hand motion (motion of the centroid during the gesture) and local motion (motion of the fingers with respect to the centroid) are most effective. Hence, we came up with a novel approach which uses both the local and global features of a gesture for Recognition.

In our method, we have used dynamic Signs of ISL as inputs to the system Also, the use of Microsoft Kinect has given depth information of our hands hence improving the accuracy of recognition. The proposed method integrates both local as well as global information of the dynamic sign. A new trajectory based feature extraction method using the concept of Axis of Least Inertia (ALI) is proposed for global feature extraction. An eigen distance based method using the seven 3D key points- (five corresponding to each finger tips, one corresponding to centre of the palm and another corresponding to lower part of palm), extracted using Kinect is proposed for local feature extraction.

May 21, 2013

II. Related Works

Most of the research works in sign language recognition is performed on sign languages other than ISL. Of recent, this area is a gaining popularity among research professionals. Computer vision and pattern recognition techniques [3], involving object detection, feature extraction, clustering, and classification, have been successfully used for many gesture recognition systems. Image-processing techniques such as analysis and detection of shape, texture, color, motion, optical flow, image enhancement, segmentation, and contour modeling, have also been found to be effective. Connectionist approaches, involving Multilayer Perception (MLP), Time Delay Neural Network (TDNN), and radial basis function network (RBFN), have been utilized in gesture recognition as well. While static gesture (pose) recognition can typically be accomplished by template matching, standard pattern recognition, and neural networks, the dynamic gesture recognition problem involves the use of techniques such as time- compressing templates, dynamic time warping, HMMs, FSM etc[4].

The earliest reported works on sign language recognition is mainly based on data glove based methods. These methods are not user friendly and are more expensive. Later some systems are developed for vision based recognition purposes. They compared the efficiency of glove based and vision based systems and found that vision based system is more efficient.[5] Many researchers conducted surveys which concentrates on the research works in vision based sign language recognition.[10] Here data acquisition, feature extraction and classification methods were employed for the analysis of sign language gestures.

Xiaodong Yang and YingLi Tians paper on Eigen Joints[2] used for activity Recognition helped us in identifying local features as the distance from the centroid to each finger tip. Yang Mingqiang, Kplama Kidiyo and Ronsin Joseph [8] have classified different methods by which a curve can be segmented and used for feature extraction. Here methods like the quadtree representation of curves and bag of features method have been discussed.Geetha M and Manjusha UC[5] have made use of B spline curves for the recognition of static hand gestures. They have made use of the Chain code method by dividing the space into eight octants and plotting the Maximum Curvature Points (MCP) in this space.

III. System Overview

We propose a system which can be used to identify and recognize the gestures based on Indian Sign Language. These gestures are dynamic and use both the hands for expression. We use Microsoft Kinect for inputting the gesture to the system. After the initial pre processing of the coordinate values, the B spline trajectory is plotted for the centroid, index, thumb, middle, ring and small finger motion. These are then subject to feature extraction that extracts the local and global features that can uniquely indentify a particular gesture. For global feature extraction, we proposed a new method using the concept of Axis of Least Inertia. (ALI). Twenty five key points are extracted from each gesture and the distance between those points to the ALI is computed. These distance vectors are taken as the global features. For local features the distance between each finger tip to centroid is computed in each frame

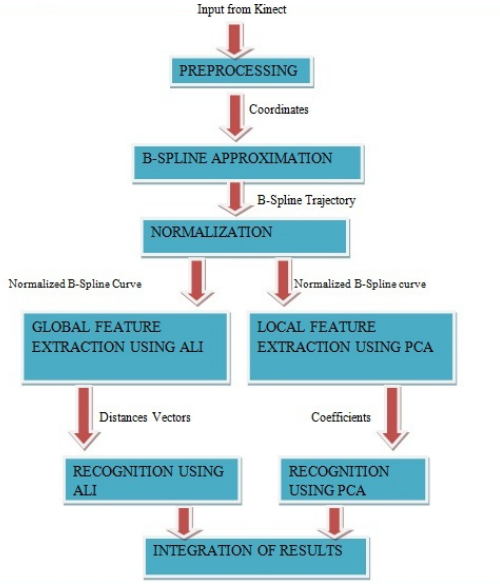

Fig. 1. Block diagram representing the structure of our system

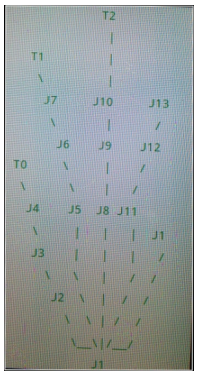

Fig. 2. Skeletal mapping of the joints of the hand through Kinect

and a matrix is created with all these distance values. This is then subject to Principle Component Analysis (PCA). An integration of both these results helps us to identify the gesture correctly.

A. Preprocessing

1) VIDEO INPUT: Microsoft Kinects 3Gear technologies have an option to visualize the hand motion as a skeletal hand motion on screen. We have utilized this option for getting the input coordinate values. The steps involved in this phase are:

1) The gesture is shown to the Kinect cameras that sense the motion of the hand in a skeletal frames
2) The program outputs the value of coordinates (x,y,z) in each frame in the following format

(Left hand) Wrist, Centre of the palm, Thumb, Index finger, Middle finger, Ring finger, Small finger

(Right hand) Wrist, Centre of the palm, Thumb, Index finger, Middle finger, Ring finger, Small finger

2) CONVERSION TO POSITIVE: The coordinates obtained from Kinects SDK can be negative. These have to be converted to positive in order to plot them onto an image.

3) Model the trajectory Using B-Spline: B-Spline is a spline function that has minimal support with respect to a given degree, smoothness, and domain partition. It is a generalization of a Bezier curve. The properties of the curve such as spatial uniqueness, boundedness and continuity, local shape controllability, and invariance to affine transformation make them an efficient choice for curve representation. Fig 5 shows a sample B-Spline curve. A closed cubic B-Spline with $n+l$ parameters $C_0, C_1, ..., C_n$, (control points) consists of $n+1$ connected curve segments $r_i(t) = (x_i(t), y_i(t))$, each of which is a linear combination of four cubic polynomials $Q_i(t)$ in the parameter t, where t is normalized for each such segment between 0 and 1 $(0 \leq t \leq 1)$, i.e.,

$$r_i(t) = C_{i-1}Q_0(t) + C_iQ_1(t) + C_{i+1}Q_2(t) + C_{i+2}Q_3(t) \tag{1}$$

The contour of the gesture is subjected to B-Spline Approximation taking the centroid as the control points

B-Spline is a good choice for the shape representation and analysis due to the following properties: (i) smoothness and continuity which allows any curve to consist of a concatenation of curve segments, yet be treated as a single unit; (ii) built-in boundedness, (iii) ease of specifying the range of a multi-valued curve; (iv) the decoupling of the x and y coordinates, with each having its parametric representation, is treated separately; (v) shape invariance under transformation (affine and projective transformations) (vi) local controllability which implies that local changes in shape are confined to the B-Spline parameters local to that change.

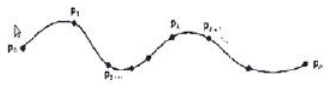

Fig. 3. B-Spline Curve

Given m real values t_j called knots with

$$t_0 \leq t_1 \leq ... < t_{m-1} \tag{2}$$

A B-Spline of degree n is a parametric curve

$$S : [t_n, t_{m-n-1}] \longrightarrow R^d \tag{3}$$

composed of a basis B-Spline $b_{i,n}$ of degree n

$$S(t) = \sum_{i=0}^{m-n-2} P_i b_{in}(t), t \in [t_n, t_{m-n-1}] \tag{4}$$

We are using a Cubic B-Spline. The B-Spline formulation for a single segment can be written as

$$S_i(t) = \sum_{k=0}^{3} P_{i-3+k} b_{i-3+k,3}(t), t \in [0,1] \tag{5}$$

Hence the basis functions are :

$$\begin{aligned} B_{i3}(t) &= \frac{1}{6} \quad if \quad i \leq t < i+1 \\ &= \frac{1}{6}[-3(t-i-1)^3 + 3(t-i-1)^2 + 3(t-i-1) \\ &+ 1] \qquad if \quad (i+1) < t < (i+2) \\ &= \frac{1}{6}[3(t-i-2)^3 - 6(t-i-2)^2 + 4] \\ &\qquad if \quad (i+2) \leq t < (i+3) \\ &= \frac{1}{6}[1-(t-i-3)]^3 \\ &\qquad if \quad (i+3) \leq t < (i+4) \\ &= \quad 0 \quad Otherwise \end{aligned} \tag{6}$$

The blending functions put in the matrix form:

$$S_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \tag{7}$$

4) Normalization: Different people may show the same gesture in different scales. Therefore, scalability can be an issue during processing. To overcome this problem, we have normalized the trajectory so that no matter how the gesture is shown, it will always fit into a 100 X 100 frame.

$$x_v = (x_{vmin} + ((x_w - x_{wmin}) * scale_x)) \tag{8}$$

$$y_v = (y_{vmin} + ((y_w - y_{wmin}) * scale_y)) \tag{9}$$

x_{vmin} is the minimum x-point in the normalized frame,x_{vmax} is the maximum x-point in the normalized frame,x_{wmin} is the minimum x-point in the non-normalized frame, x_{wmax} is the maximum x-point in the non-normalized frame,y_{vmin} is the minimum y-point in the normalized frame,y_{vmax} is the maximum y-point in the normalized frame,y_{wmin} is the minimum y-point in the non-normalized frame,y_{wmax} is the maximum y-point in the non-normalized frame,x_v and y_v are the Non normalized coordinates in the B spline trajectory.

Data: Bspline points to be normalized
Result: Normalization of B spline curve
initialization;
$scale_x \longleftarrow (x_{vmax} - x_{vmin})/(x_{wmax} - x_{wmin})$;
$scale_y \longleftarrow (y_{vmax} - y_{vmin})/(y_{wmax} - y_{wmin})$;
while *For all points* **do**
  $xv \longleftarrow (int)(x_{vmin} + ((x_w - x_{wmin}) * scale_x))$;
  $yv \longleftarrow (int)(y_{vmin} + ((y_w - y_{wmin}) * scale_y))$;
end

Algorithm 1: Algorithm for Normalization

B. FEATURE EXTRACTION

In order to recognize gestures based on the input values, feature vectors play a commandalble part. This makes the extraction of feature vectors a.k.a Feature Extraction extremely important in any gesture recognition related work. Our method takes into account both the local and global features associated with a gesture. Global Feature Vectors , are extracted on the basis of the centroid of the hand, while local feature vectors are extracted according to the finger movements.

Fig. 4. 25 Global features of the word $Valley$

1) GLOBAL FEATURE EXTRACTION:

The new method proposed is based on the concept of Axis of least Inertia.

a) AXIS OF LEAST INERTIA:

The axis of least inertia (ALI) of a shape is defined as the line for which the integral of the square of the distances to points on the shape boundary is a minimum. Once the ALI is calculated, each point on the shape curve is projected on to ALI. The two farthest projected points say E1 and E2 on ALI are chosen as the extreme points as shown in Figure 5. The Euclidean distance between these two extreme points defines the length of ALI.

Data: TO FIND THE GLOBAL FEATURES
Result: GLOBAL FEATURE VECTORS
1.Find the key points in the Bspline Trajectory;
2. Using clustering reduce the number of Key points;
3. Take into account a fixed number of other points excluding the key points;
4. Get a fixed number of selected points (key points + other points);
while *For all selected points* **do**
  Find the distance between these selected points to the ALI.;
end
5. The resultant distance vectors that are considered as the global features.;

Algorithm 2: Algorithm for Extraction of GlobalFeature Vectors

2) Local Feature Extraction: An eigen distance based method using the seven 3D key points- (five corresponding to each finger tips, one corresponding to centre of the palm (centroid) and another corresponding to lower part of palm), extracted using Kinect is proposed for local feature extraction. For every frame, distance from tip of each finger to the centroid is calculated. Principal Component Analysis is applied to the distance measures to reduce the dimensionality giving rise to eigen distances. and stored as a matrix. This distance is computed in each frame. It is stored in a matrix in this order:
Gesture1 Gesture2 Gesture 3 . .. . Gesture n
D(t,c,f1) D(t,c,f1) D(t,c,f1) .. . . D(t,c,fn)
D(t,c,f2) D(t,c,f2) D(t,c,f2) . . D(t,c,fn)
D(t,c,f3) D(t,c,f3) D(t,c,f3) . ... D(t,c,fn)
.
.
D(t,c,fn) D(t,c,fn) D(t,c,fn) .. .. D(t,c,fn)
This is considered as the input matrix to PCA.(Principle Component Analysis)

a) PRINCIPAL COMPONENT ANALYSIS: Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.

STEPS IN PCA

1) 1. From the input matrix, find the covariance matrix.
2) 2. From the covariance matrix, find the Eigen values and Eigen vectors
3) 3. Select the Eigen vectors to be multiplied with input matrix based on the Eigen values
4) 4. Multiply the input matrix with the selected Eigen Vectors and get coefficients

Data: TO FIND LOCAL FEATURES USING PCA
Result: LOCAL FEATURE VECTORS
while *From all fingertips* **do**
  $Input_Matrix \longleftarrow Distance(fingertip, centroid)$;
end
2. Do $Resultant_Matrix \longleftarrow Input_Matrix - Mean$;
3.$Covariance_Matrix \longleftarrow Resultant_Matrix/(N-1)$;
4. From the covariance matrix, find Eigen values and Eigen vectors;
5. Sort the Eigen values. Find the number of Eigen values to be considered using;
for *i=1* **to** *N* **do** $Val \longleftarrow sum(i)/Totalsum$;
if *Val* $>$ *0.95* **then** Choose i as the number of points required;
6. Take the corresponding Eigen vectors and multiply it with the transpose of the input matrix.;
7. This results in the coefficients ;

Algorithm 3: Algorithm for finding Local Features using PCA

IV. RECOGNITION

A. Glocal Feature Extraction

In the Global Feature Extraction, we had 25 distance vectors and 15 coefficients corresponding to each gesture. When any new gesture arrives, its feature vectors

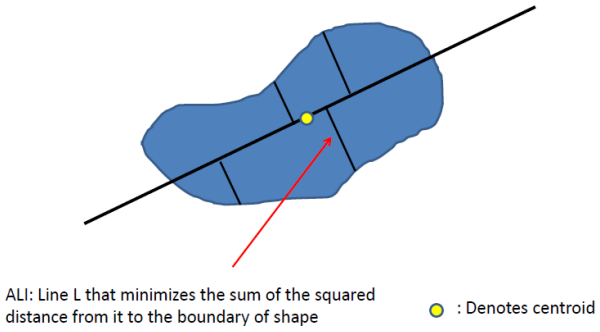

Fig. 5. An example ALI for a shape

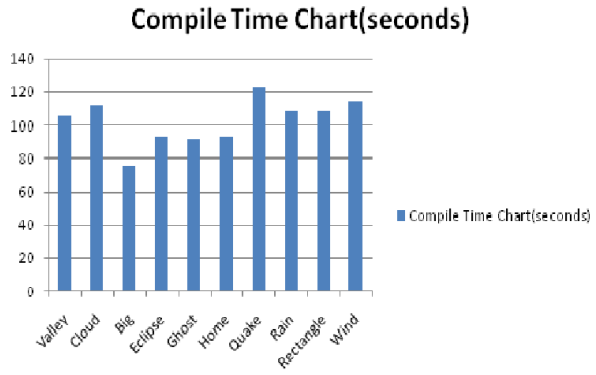

Fig. 6. Time Complexity

are compared to the Feature Vectors present using Euclidean Distance. The gesture corresponding to which we get minimum Euclidean Distance is given as the recognized gesture.

B. Local Feature Extraction

Corresponding to each finger-centroid we have 15 coefficient values. This gives us a total of 75 global feature vectors. To make it simpler, we considered the feature vectors of thumb, middle finger and small finger alone. Thus reducing the number of feature vectors to 45.

Any new gesture is compared to these feature vectors. Euclidean distance of the feature vectors is found and the gesture with minimum Euclidean distance is given as the output recognized gesture. Now recognized gestures from local features and global features are integrated to output the final result.

V. TESTING

A. Unit Testing

Under unit testing, we test separate modules of the system. In our case, we tested if the output of each phase was satisfactory. Starting with pre processing, we checked if all the coordinates are converted to positive correctly. Then we plotted the B spline trajectory and checked if it resembled the gesture shown to the system. Even after normalizing, we plotted the trajectory to see if the normalized trajectory was in the 100 x 100 frame. Next, after extracting the local and global features, we tested the results separately and looked at the recognition rate of these individually. For each gesture, we have taken the following samples as data sets for testing. Two Good samples Two Average samples Two Bad Samples Out of the six, 1 good sample is taken as a reference For the good sample that is taken as reference, 25 global features are stored for each gesture. Therefore, we obtain distance vectors corresponding to each gesture. The other 5 samples of each gesture are given as input and tested for results. A confusion matrix is then created showing the accuracy of recognition. For the good sample that is taken as reference, 15 local features are stored for each finger of a gesture. For simplicity, we considered the thumb, middle and small finger distance vectors alone. This is then given to PCA to get the coefficients. Therefore, we obtain coefficients corresponding to each gesture.

The other 5 samples of each gesture are given as input and tested for results. A confusion matrix is then created showing the accuracy of recognition

B. Integration Testing

The outputs using local features and global features are integrated to produce the final result. We have also implemented the system on another system and made sure it is working correctly. This is to make sure that there is no confusion in the application while it is running on different machines. Making sure that all the data is saved correctly and there is no loss of the captured data is also an important phase of testing. We started from acquiring the input coordinates and made sure that all arrays and matrices are filled with the correct coordinate values. Since the size of the matrix was huge, making sure that the data is stored correctly was a tedious process which needed attention.

C. Validation Testing

The outputs obtained are compared to the results that must be derived. This way we check for validation of the system. In case of a incorrectly recognized gesture we make a deficiency list specifying which gesture is misinterpreted and in what deviation. This helps us understand the accuracy of recognition better.

D. Performance Testing

Performance bounds are set during the design stage of the software development. These bounds help us in identifying the effectiveness of the software.

VI. RESULTS

We conducted our experiments with 6 samples (2 best cases , 2 average cases , 2 worst cases) of each word.The whole system is implemented with the help of OpenCV library in Ubuntu 12.04.

A. Experimental Analysis

Figure 7 indicates the time complexity of each gesture.

Figure 8 indicates the Accuracy of Global Feature vectors. Note that the best case for each gesture give us an accuracy of

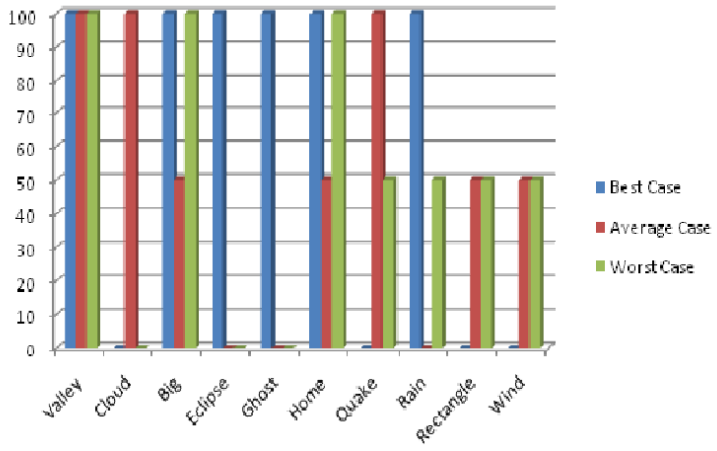

Fig. 7. Accuracy of Global feature vectors

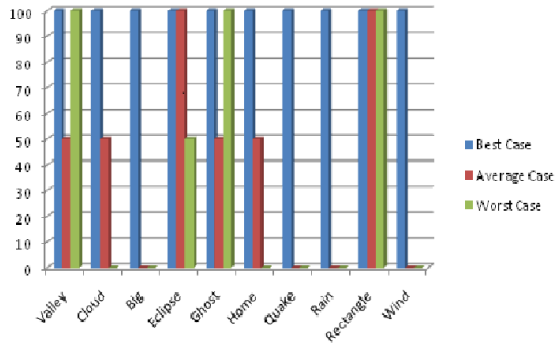

Fig. 8. Accuracy of local feature vectors

100%. From the above, we can see that the accuracy of long gestures like cloud and eclipse is comparatively low. Therefore there is a need to include the local features for better results.

Figure 9 indicates the local feature vectors. Best case for each gesture recorded an 100% for local feature extraction also. Figure 10 indicates the integrated feature vectors, that is the result combining both global and local features .Best case for each gesture recorded an 100% for integrated feature extraction also.
From the above graphs we can see that the accuracy obtained for all gestures given in the best case is 100% . Average Complexity has an accuracy of 40% and worst case a accuracy of 25%. It is to be noted that disturbances in the trajectory (jagged motion) can influence the accuracy of recognition.

B. Performance Analysis

As the system is under research 100% accuracy cannot be guaranteed. The results are also influenced by external factors such as uniformity in motion of the trajectory. This is very difficult using the Kinect 3 Gear cameras since there is lot of jagged motion of fingers. However, keeping in mind this constraint, the implemented method has achieved 100

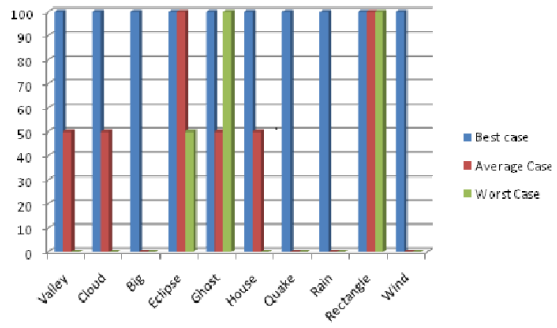

Fig. 9. Accuracy of Integrated(Global + local) feature vectors

VII. CONCLUSION

The proposed gesture recognition system can handle different types of words in a common vision based platform. Our approach uses both local and global features for recognition which improves the accuracy of recognition. So our system is a promising approach in this field. The system is suitable for complex ISL dynamic signs. However, it is to be noted that the proposed gesture recognizer cannot be considered as a complete sign language recognizer, as for complete recognition of sign language, information about other body parts i.e., head, arm, facial expression etc are essential. The experimental results show that the system is sufficient to claim a ”working system” for native Indian sign language recognition. The system is designed to support recognition of words in ISL. This can be enhanced to recognize continuous sentences as well.

ACKNOWLEDGMENT

We would like to thank Mr.Akshay and Mr. Unnikrishnan R, SAVE LABS,Amritapuri,Kollam, for their support. We also take this opportunity to thank all the faculties and staff of Amrita School of Engineering, Amritapuri Campus for their valuable help.Above all, we render our gratitude to God Almighty, who bestowed upon his blessings and instilled upon us self-confidence and strength which helped to complete the project.

REFERENCES

[1] T. Starner and A. Pentland, ”Real-time american sign language recognition from video using hidden markov models”, Technical Report, M.I.T Media Laboratory Perceptual Computing Section, Technical Report No. 375, 1995.

[2] Xiaodong Yang and YingLi Tian , Eigen Joints Based Action Recognition Using Nave Bayes Nearest Neighbour, THE CITY COLLEGE OF NEW YORK, NEW YORK.

[3] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis.New York: Wiley, 1973.

[4] Sushmita Mitra, Senior Member, IEEE, and Tinku Acharya, Senior Member, IEEE, Gesture Recognition: A Survey, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART C: APPLICATIONS AND REVIEWS, VOL. 37, NO. 3, MAY 2007.

[5] Geetha M and Manjusha U C , A vision based Recogniton of Indian Sign language Alphabets and Numerals using Bspline approximation, INTERNATIONAL JOURNAL ON COMPUTER SCIENCE AND ENGINEERING(IJCSE), VOL. 4, NO. 3, MARCH 2012.

[6] Matthew Tang, Recognising hand gestures with Microsoft Kinect, STANFORD UNIVERSITY.

[7] Rafiqul Zaman Khan and Noor Adnan Ibraheem, COMPARITIVE STUDY OF HAND GESTURE RECOGNITION SYSTEM , A. M. U , Aligarh, India

[8] Yang Mingqiang,Kplama Kidiyo and Ronsin Joseph , A survey of shape feature extraction Techniques, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART C: APPLICATIONS AND REVIEWS, VOL. 30, NO. 2, MAY 2000.

[9] Mu-Chun Su , A Fuzzy Rule Based Approach to Spatio Temporal Hand Gesture Recognition, PATTERN RECOGNITION, PENG-YENG YIN, VERSION. 1, NO. 43-90, 2008

[10] Sylvie C.W. Ong and Surendra Ranganath , Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE