A Vision Based Recognition of Indian Sign Language Alphabets and Numerals Using B-Spline Approximation

Geetha M
Department of Computer Science and Engineering
Amrita Vishwa Vidyapeetham
Amritapuri
geetham@am.amrita.edu

Manjusha U C
Department of Computer Science and Engineering,
Amrita Vishwa Vidyapeetham,
Amritapuri
sree01manju@gmail.com

Abstract - Sign language is the most natural way of expression for the deaf community. The urge to support the integration of deaf people into the hearing society made the automatic sign language recognition, an area of interest for the researchers. Indian Sign Language (ISL) is a visual-spatial language which provides linguistic information using hands, arms, facial expressions, and head/body postures. In this paper we propose a novel vision-based recognition of Indian Sign Language Alphabets and Numerals using B-Spline Approximation. Gestures of ISL alphabets are complex since it involves the gestures of both the hands together. Our algorithm approximates the boundary extracted from the Region of Interest, to a B-Spline curve by taking the Maximum Curvature Points (MCPs) as the Control points. Then the B-Spline curve is subjected to iterations for smoothening resulting in the extraction of Key Maximum Curvature points (KMCPs), which are the key contributors of the gesture shape. Hence a translation & scale invariant feature vector is obtained from the spatial locations of the KMCPs in the 8 Octant Regions of the 2D Space which is given for classification.

Keywords – Indian Sign Language (ISL), Maximum Curvature Points (MCPs), Gesture recognition, B-Spline Approximation, Parametric Continuity, Threshold Iteration.

I. INTRODUCTION

Sign language is a movement language which expresses certain semantic information through series of hands and arms motion, facial expressions & head/body postures. It is the basic communication medium between the deaf people. A translator is usually needed when an ordinary person wants to communicate with a deaf one. Deaf people use sign language as their medium of communication. Sign language recognition is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and psychology. It is a comprehensive problem because of the complexity of the visual analysis of hand gestures and the highly structured nature of sign languages. It is a very important area not only from engineering point of view but also for its impact on the society. A functioning sign language recognition system can provide an opportunity for a mute person to communicate with non-signing people without the need for an interpreter. Every country has its own sign language. There are different sign languages all over the world such as American Sign Language (ASL), British Sign Language (BSL), French Sign Language, Danish Sign Language, Taiwan Sign Language, Australian Sign Language, etc. All these languages were developed independently. There is no unique way in which such recognition can be formalized. Sign languages are well structured languages with a phonology, morphology, syntax and grammar distinctive from spoken languages.

In the same way, Indian Sign Language was also developed for Indian deaf community. It differs in the syntax, phonology, morphology and grammar from other country's sign languages. ISL uses static and dynamic hand gestures, facial expressions, head/body postures, locations of hand with respect to body etc. to represent signs. It is more challenging than other sign languages due to the following reasons:

i. Most of the signs make use of the gestures of both the hands together.
ii. One hand moves faster than the other at times.
iii. Many of the gestures result in occlusions.
iv. Complicated hand shapes.
v. Locations of the hand with respect to body contribute to the Sign.
vi. Hand contacts the body for some signs.
vii. Involves both global and local hand motion.

In general, the entire system of the vision-based sign language recognition is simpler than the data glove based approach which uses special input devices for tracking and digitizing hand and finger motions into multi-parametric data. These devices are too expensive and the users might feel uncomfortable when they communicate with a machine. Without specialized tracking devices, vision based system helps to reliably detect and track the gestures.

Since ISL got standardized only recently and also since tutorials on ISL gestures were not available until recently, there are very few research work that has happened in ISL recognition. Taking into consideration the challenges in ISL gesture recognition, we are proposing a novel method for recognition of Indian sign language alphabets and numerals using B-Spline approximation. Our algorithm approximates the boundary extracted from the Region of Interest, to a B-Spline curve by taking the Maximum Curvature Points (MCPs) as the control points. Then the B-Spline curve is subjected to iterations for smoothening resulting in the extraction of Key Maximum Curvature points (KMCPs), which are the key contributors of the gesture shape. The spatial locations of the KMCPs in the 8 Octant Regions of the 2D space is considered as the feature vector for classification.

II. RELATED WORKS

The earliest reported work on sign language recognition is available in [1] in which Starner and Pentland developed a glove-environment system capable of recognizing a subset of the American Sign Language (ASL).Sushmita Mitra and Tinku Acharya had provided a survey on gesture recognition, with particular emphasis on hand gestures and facial expressions[8]. Deng-Yuan Huang1, Wu-Chih Hu2, Sung-Hsiang Chang conducted a work [9] which explains the use of gabor filters to acquire desirable hand gesture features. The principal components analysis (PCA) method is then used to reduce the dimensionality of the feature space.

The work presented by Sara Bilal, Rini Akmeliawati, Momoh Jimoh El Salami, Amir A. Shafie [10] goals to develop a system for automatic translation of static as well as dynamic gestures of Indian Sign Language. One prominent approach describes the vision based recognition technique [18] to achieve visual information in the form of feature vector. For the extraction of feature vectors for recognition, several different techniques were used till now. A number of approaches have been proposed for curve modeling such as Fourier descriptors, chain codes, polygonal approximation, curvature primal sketch, medial axis transform, autoregressive models, moments, parametric algebraic curves, curvature invariant, stochastic transformation, implicit polynomial functions, bounded polynomials, B-Splines, reaction diffusion etc. Fourier Descriptor (FD) is one of the well known, relatively simple methods of shape matching. In FD, contour of the shape is transformed to frequency domain to perform the comparison. Generally FD method is immune to rotation or scaling of the shape and noise. However FD is prone to varying phase of frequency components [12].

Wavelet Descriptors (WD) is an advanced version of the FD. In WD, shape in spatial domain is converted to a spatial-frequency domain. Thus WD has frequency details specific to the spatial location. The drawback of WD is its high dependency on contour starting point for successful matching. Another work [13] used B-Spline with curvature-based, knot points-based and residual error based matching techniques to match shape contours, which generally performs well in basic shape matching. Nuwan Gamage, Rini Akmeliawat conducted a study [11] which mentions static hand recognition using linear projection methods which is little more advantageous when compared to other methods. Paper [14] introduces a method that combines the advantages of B-Spline that are continuous curve representation and the robustness of CSS matching with respect to noise and affine transformation.Their experimental results showed the robustness and accuracy of the proposed method in B-Spline curve matching.

III. SYSTEM OVERVIEW

The problem discussed in this paper is a vision based identification of the static signs of Indian sign language (ISL) alphabets and numerals. The signs considered for recognition includes the alphabets (A-Z) excluding J and H, and numerals (0-9). The system deals with images of bare hands, which allows the user to interact with the system in a natural way and in whatever environment he is comfortable with. A static sign is determined by certain configuration of the hand, while a dynamic sign comprises of one or more moving gesture *i.e.* a sequence of hand movements and configurations. As an initial phase, our algorithm focuses on static signs for the alphabets and numbers. The recognition of the alphabets j and h (dynamic signs) requires additional steps for identification which traces the trajectory of motion and extracts the shape of the trajectory. The algorithm considers hand region as the area of interest. We are assuming that no other objects other than the Region of Interest are present in the background. Fig 1 shows the dataset we used.

Fig 1: ISL Dataset

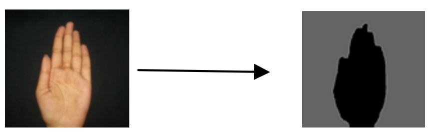
We propose a new method which uses B-Spline approximation for the shape matching of static gestures of ISL alphabets & numerals. The hand shape is approximated to a B-Spline curve. The B-Splines are piecewise polynomial functions that provide local approximations to contours/surfaces using a small number of control points. The advantage of the B-Spline curve, which is continuous curve representation, is applied here. Fig 2 shows the overall system design. The sign gesture is boundary traced to obtain the Maximum curvature points (MCPs).The extracted MCPs are considered as the control points. The control points are then fitted with piecewise continuous parametric polynomial functions called B-Spline curves. The approximated B-Spline curve is subjected to a set of iterations for the purpose of smoothening. The maximum curvature points retained after the series of smoothening called the Key Maximum Curvature Points (KMCPs) are the major contributors for the shape of the gesture. The spatial location of those key points are extracted and considered as the feature vector. For this purpose the entire 2D space of the gesture is divided into 8 octants. The number of key points which falls in each of the octant is counted. Hence our feature vector includes a set of 8 values each corresponds to the count of the KMCPs in each of the eight octants. Finally recognition is done using SVM classifier.

The method comprises of the following steps:

- Preprocessing the image input.
- Boundary Tracing
- Find the MCPs
- Boundary approximation to a B-Spline curve.
- Resampling and Smoothening
- Feature extraction.
- Classification and recognition.

A. Preprocessing

Image preprocessing includes the set of operations on images whose goal is the improvement of the image data that suppresses undesired distortions or enhances some image features important for further processing. Hand region is extracted from the image using various image processing techniques like image cropping, connected component analysis, background subtraction, edge detection etc.
Fig 3(a) shows an image of number 5 in ISL, Fig 3(b) shows the image after preprocessing.



(a) (b)
Fig 3: (a) Actual image (b) Preprocessed image

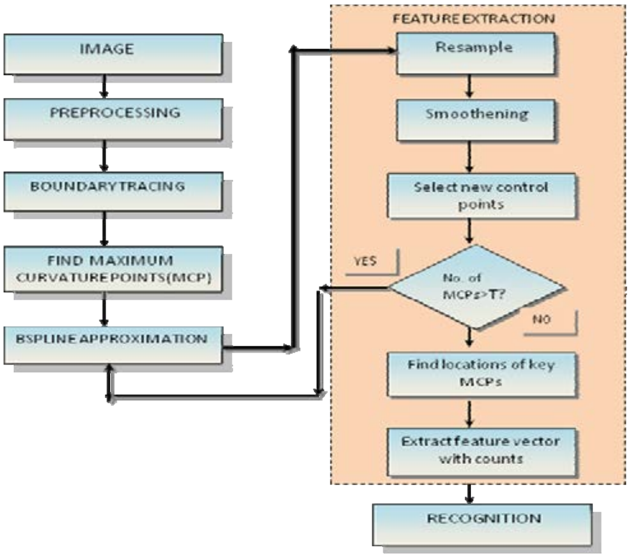


Fig 2: System design

B. Boundary Tracing

The boundary of the preprocessed gesture is extracted using the algorithm for boundary tracing.
Algorithm1 (Boundary Tracing)

1. We define a variable dir which stores the direction of the previous move along the border from previous border element to the current border element. We assign
   a) dir=0 if the border is detected in 4-connectivity.
   b) dir=7 if the border is detected in 8-connectivity.
2. Search the 3×3 neighborhood of the current pixel in an anti-clockwise direction, beginning the neighborhood search in the pixel positioned in the direction.
   a) (dir + 3) mod 4
   b) (dir + 7) mod 8 if dir is even
      (dir + 6) mod 8 if dir is odd

   The first pixel found with the same value as the current pixel is a new boundary element P_n.. Update the dir value.
3. If the current boundary element P_n is equal to the second border element P_1, and if the previous border element P_{n-1} is equal to P_0. Otherwise repeat step (2).
4. The detected pixel borders are represented by the pixels P_0……P_{n-2}.

C. Finding the Maximum Curvature points (MCPs)

MCPs are selected from the traced boundary by considering the neighboring pixels having the degree of curvature greater than a threshold. This is done by taking the angular difference at each position on the curve. Fig 4 shows an example. We take the tangent value of the angle made by the line segment connecting the neighboring pixels with the horizontal which gives the degree of curvature at that particular region. A maximum curvature point occurs whenever there is a local maximum in the angular change.

$$\tan\theta = \frac{\Delta y}{\Delta x} \qquad (1)$$

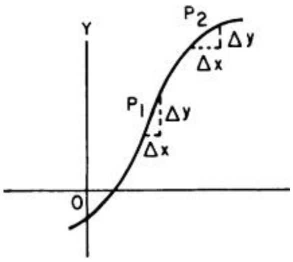


Fig 4: Example for tangent calculation

Algorithm 2 MCP Selection (Threshold T)

For each pair of neighboring pixels (P1P2) which are at a reasonable distance,

a. Compute the curvature at P1 and P2 using the equation,

$$\tan\,\theta = \frac{\Delta y}{\Delta x}$$

b. Consider only those pixels with maximum angular change (i.e. maximum difference between two angles).
Save the points whose (Difference in tan θ) > T

The MCPs selected are given as the control points for B-Spline approximation.

D. B-Spline Approximation

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 & 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.,

$$r_i\,(t) = C_{i-1}\,Q_0\,(t) + C_i\,Q_1\,(t) + C_{i+1}\,Q_2\,(t) + C_{i+2}\,Q_3\,(t) \qquad (2)$$

The properties which make B-Splines suitable for the shape representation and analysis are the following: (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 5: B-Spline curve

Given m real values t_j called knots with

$$t_0 \leq t_1 \leq \cdots \leq t_{m-1} \qquad (3)$$

a B-Spline of degree n is a parametric curve

$$S : [t_n, t_{m-n-1}] \to R^d \qquad (4)$$

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

$$S(t) = \sum_{i=0}^{m-n-2} P_i b_{i,n}(t), t \in [t_n, t_{m-n-1}] \quad (5)$$

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

$$S_i(t) = \sum_{k=0}^{3} P_{i-3+k}\, b_{i-3+k,3}(t), t \in [0,1] \quad (6)$$

Hence the basis functions are :

$$B_{i3}(t) = \begin{cases} \frac{1}{6}(t-i)^3 & \text{if } i \le t < i+1 \\ \frac{1}{6}[-3(t-i-1)^3 + 3(t-i-1)^2 + 3(t-i-1) + 1] & \text{if } i+1 \le t < i+2 \\ \frac{1}{6}[3(t-i-2)^3 - 6(t-i-2)^2 + 4] & \text{if } i+2 \le t < i+3 \\ \frac{1}{6}[1-(t-i-3)]^3 & \text{if } i+3 \le t < i+4 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The blending functions put in the matrix form:

$$S_i(t) = [t^3 \quad t^2 \quad t \quad 1]\frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}\begin{bmatrix} P_{i-1} \\ P_i \\ P_{i+1} \\ P_{i+2} \end{bmatrix} \quad (8)$$

In this system, the object contour is assumed to be approximated and presented by the B-Spline. At first from boundary pixels, the points of maximum curvature (MCPs) are determined. A maximum curvature point occurs whenever there is a local maximum angular change. These maximum curvature points are considered as the control point for the B-Spline. Finally, the B-Spline segments are exported in the form of an ordered list of control points, around the full perimeter of the contour. In this method, the number of segments generated will be 3 less than number of control points. This happens because control points are shared by neighboring segments.

E. Resampling and Smoothening

The number of boundary points of the initial B-Spline will be very large. Resampling is done on the B-Spline curve inorder to reduce the total number of boundary points from the initial B-Spline. From this resampled points, control points are selected to construct the smoothened B-spline. The sampling process not only normalizes the sizes of shapes but also has the effect of smoothening the shape. This eliminates the noise in the boundary shape as well as irrelevant shape details along the boundary. By varying the number of sampled points, the accuracy of the shape representation can be adjusted.

Resampled Count Q = M ÷ N (9)

where M= *Required number of points for each iteration.*

N=*Total number. of points.*

A set of iterations of smoothening is done on the initial B-Spline curve inorder to get the KMCPs. In each iteration we recalculate the Maximum Curvature Points from the B-Spline curve. This is done by considering those points on the curve with local maximum of the change in first order parametric derivative $\frac{d^n s}{dt^n}$, n=1 at the neighboring pixel positions. The number of iterations for smoothening (Threshold Iteration) is fixed as the iteration number, where the number of control points becomes equal to a threshold value. We fixed the number of control points as 15.Fig 6 shows spline curves for the gesture of number 5, obtained after smoothening.

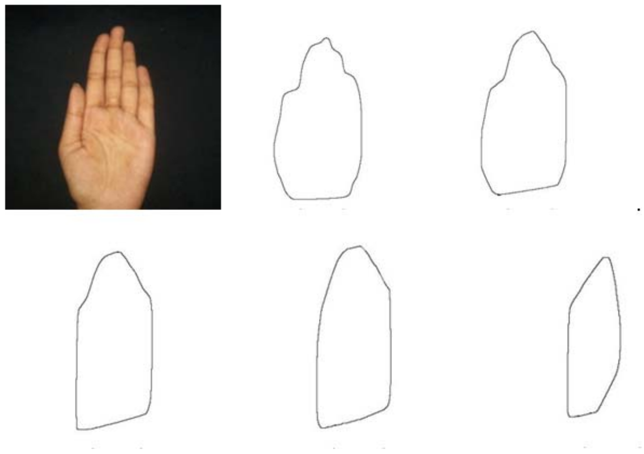


Fig 6: smoothening steps

D. Feature vector selection

For the recognition step we need to extract the feature vector which is capable of uniquely identifying all the gestures corresponding to each sign. We are proposing a new method for feature extraction wherein the area covered by the B-Spline curve corresponding to the Threshold Iteration is divided spatially into 8 octants and the KMCPs count at each spatial Octant is extracted. Threshold Iteration (TI) is chosen as that iteration where the number of control points reduces to a constant T. There we stop the smoothening. Value of T cannot be too large since we may get large no of control point many of which may not contribute much to the actual shape of the gesture. T cannot be too small also since it may result in too few control points, If some of them happened to be outliers it may completely distort the shape information.. Hence we chose the value of T as 15.The Maximum Curvature points (MCPs) corresponding to Threshold Iteration is called Key Maximum Curvature Points (KMCPs). KMCPs which are retained after the series of smoothening steps are the major contributors towards the shape of the gesture. Hence spatial locations of KMCPs will help in uniquely identifying the gesture. [FV_1, FV_2,…...FV_8] where FV_i indicates the count of KMCPs in i^{th} Octant is taken as the feature vector. Hence an eight valued feature vector is obtained, which is provided as input to the recognizer. Fig 7 shows an example of the count of alphabet A and number 5 in 8 octants of 2D space. This feature vector ensures translation invariance since we are focusing only on the Region of Interest. It also ensures scale invariance since we take the Octant Region with centre as the centre of the gesture.

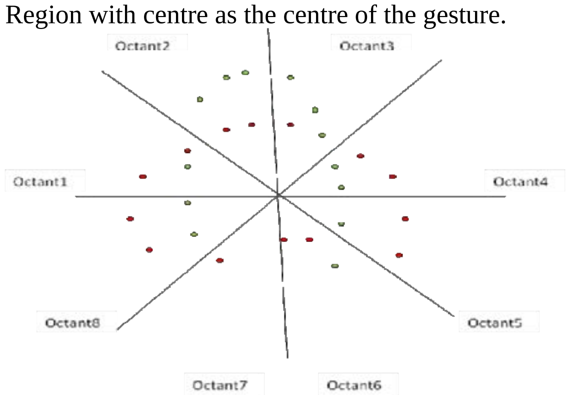


Fig 7: Dividing 2D space into octants (For example, Red dots indicate points correspond to alphabet A and Green dots indicate points correspond to number 5).

Algorithm 3: ReSampling & Smoothening

Start

//Smoothening//

1. Resample the boundary points of the B-Spline.
2. Find the maximum curvature points.
3. Construct the B-Spline with MCPs of step 2 as the control points.
4. Repeat steps 1, 2, 3 until TI is reached.
5. Calculate the count, FV_i of KMCPs in the i^{th} octant
6. Save the Feature Vector [FV_1,FV_2…FV_8]

End

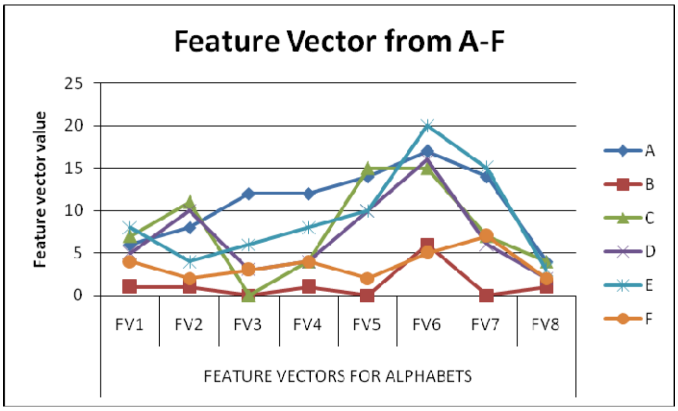
IV. RECOGNITION

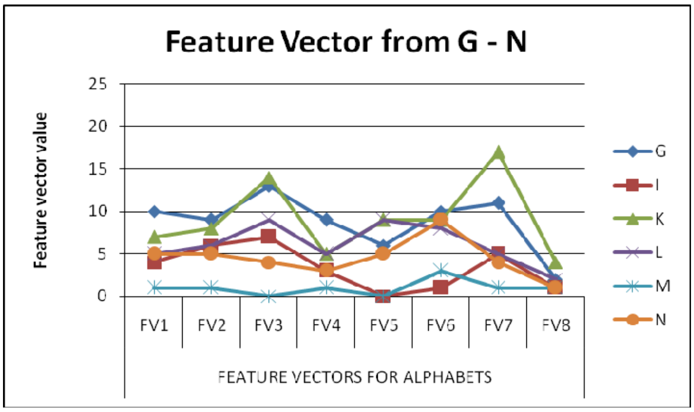
A. *Support Vector Machines*

Support Vector Machines (SVMs) are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. There are roughly three ways to solve the problem: one against-all, pairwise, and simultaneous classifications. In one-against-all classification, instead of discrete decision functions, Vapnik proposed to use continuous decision functions. In pairwise classification, the n-class problem is converted into $n(n-1)/2$ two-class problems[17]. In simultaneous formulation we need to determine all the decision functions at once, which results in simultaneously solving a problem with larger number of variables than the above mentioned methods. Here we used pair-wise separation method for classifying the instances. Since we have 26 alphabets, we have taken 26 classes for recognition and additional classes for numericals.

V. EXPERIMENTS AND RESULTS

We conducted our experiments with 50 samples of each alphabet from A - Z and numbers from 0-5. Initially the system is trained using 10 samples of each alphabet and number. The whole system is implemented with the help of OpenCV library in Microsoft Visual Studio 2010 express edition. The graphs shown below in Fig 8 indicate the value of the 8 element feature vector extracted for each member of the data set of alphabets from A-Z. As can be seen from the graphs, the shape of the curve corresponding to each alphabet is different. This clearly shows that the values of the feature vector selected are unique for each data set. Hence our algorithm is efficient enough to uniquely identify each alphabet.





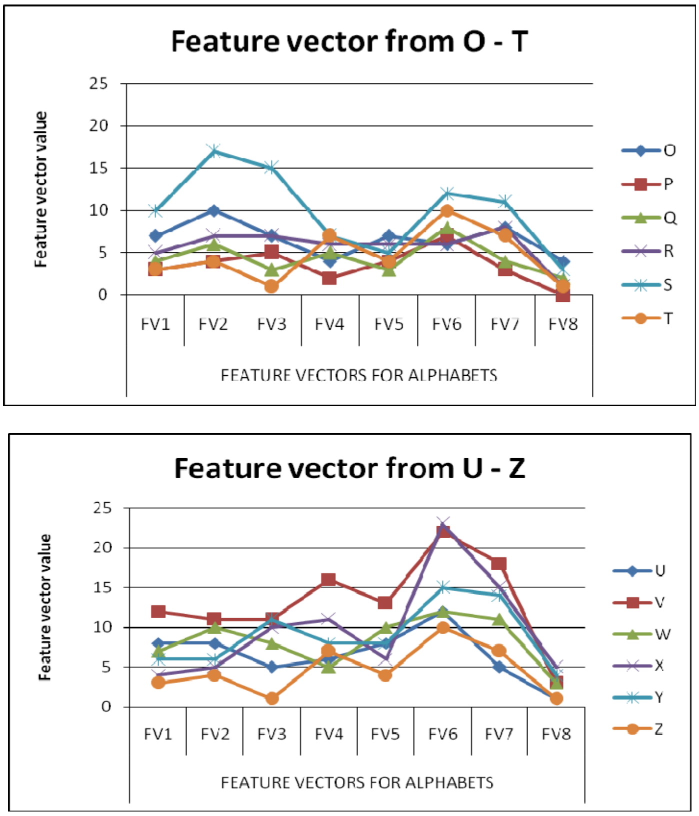


Fig 8 :Graphs showing feature vector corresponds to alphabets(A-Z) and numbers(1-5).

Recognition rates obtained are as follows in Fig 9.
Percentage Accuracy = No: of correctly classified samples ÷ Total No: of Samples

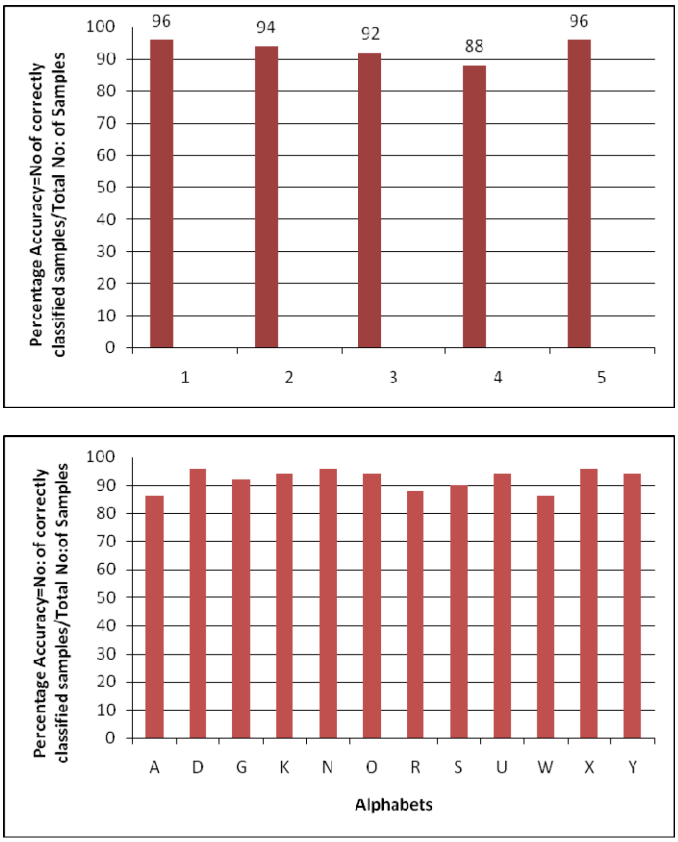


Fig 9: Graphs showing the accuracy rates of alphabets and numbers.

Our recognizer gives reasonably good recognition rates for both alphabets and numerals. A very few papers have already worked on ISL recognition. Our method is more advantageous when compared with the existing methods as it uses B-Spline curve to approximate the boundary which gives better recognition for the complex shapes in ISL.

VI. CONCLUSION AND FUTURE WORKS

The proposed gesture recognition system can handle different types of alphabets and number signs in a common vision based platform. The system is suitable for complex ISL static 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 alphabet & numeral recognition. Thus, the proposed approach will be useful and will have a sufficient amount of accuracy to recognize a hand sign gesture. Our system can be extended to include the vision based recognition of dynamic gestures corresponding to words or sentences in ISL.

VII. 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] M. Vasishta, J. Woodward and S. DeSantis. An Introduction to Indian Sign Language, All India Federation of the Deaf (Third Edition).1998.

[3] M. Vasishta, J. Woodward and K. Wilson. Sign language in India: regional variation within the deaf population. Indian Journal of Applied Linguistics 4(2): 66-74,1978.

[4] Zeshan U., Madan M. Vasishta, Sethna M. 2004, “implementation of indian sign language in educational settings”- Volume 15, Number 2, Asia Pacific Disability Rehabilitation Journal, , pp. 15-35.

[5] Ying Wu and Thomas S Huang, "Vision-based gesture recognition: A review,"LNCS: Gesture-Based Communication in Human-Computer Interaction: International Gesture Workshop, vol. 1739, pp. 103, 2004.

[6] Mulder, "Hand gestures for HCL," Technical Report 96-1, vol. Simon Fraster University, 1996.

[7] F. Quek, "Towards a vision-based hand gesture interface," in Proceedings of Virtual Reality Software and Technology, Singapore, 1994, pp. 17-31.

[8] Sushmita Mitra, Senior Member, IEEE, and Tinku Acharya, Senior Member, IEEE, “Gesture Recognition: A Survey”, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 37, NO. 3, MAY 2007.

[9] Deng-Yuan Huang1, Wu-Chih Hu2, Sung-Hsiang Chang , “Vision-based Hand Gesture Recognition Using PCA + Gabor Filters and SVM” ,2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing.

[10] Sara Bilal1, Rini Akmeliawati2, Momoh Jimoh El Salami ,Amir A. Shafie, “Vision-based Hand Posture Detection and Recognition for Sign Language-A study”, 2011 4th International Conference on Mechatronics (ICOM), 17-19 May 2011, Kuala Lumpur, Malaysia.

[11] Nuwan Gamage, Rini Akmeliawat, “Static Hand Sign Recognition using Linear Projection Methods”, Proceedings of the 4th International Conference on Autonomous Robots and Agents, Feb 10-12, 2009, Wellington, New Zealand.

[12] Wang and J. A. Bangham, "PCA Based Shape Descriptors for Shape Retrieval and The Evaluations," in International Conference on Computational Intelligence and Security. vol. 2: IEEE, 2006, pp. 1401 - 1406.

[13] F. S. Cohen, Z. Huang, and Z. Yang, "Curve Recognition Using B-spline Representation," IEEE, 1992.

[14] Yue Wang, Member, IEEE, and Eam Khwang Teoh, Member, IEEE , “2D Affine-Invariant Contour Matching Using B-Spline Model”, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 29, NO. 10, OCTOBER 2007.

[15] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin,“A Practical Guide to Support Vector Classifcation”, Department of Computer Science National Taiwan University, Taipei 106, Taiwan.

[16] Introduction to Machine Learning, Second Edition, Ethem Alpaydin, MIT Press Cambridge, Massachusetts, London, England.

[17] Isaac Garcia Incertis, Jaime Gomez Garcia-Bermejo, Eduardo Zalama Casanova, "Hand Gesture Recognition for Deaf People Interfacing", icpr, 18th International Conference on Pattern Recognition (ICPR'06), Vol. 2, pp. 100-103, 2006.

[18] Thomas Coogan , George Awad, Junwei Han and Alistair Sutherland, “Real time hand gesture recognition including hand segmentation and tracking”, ISVC 2006 - 2nd International Symposium on Visual Computing, 6-8 November 2006, Lake Tahoe, NV, USA. ISBN 978- 3-540-48628-2.