International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:8, 2009

Posture Recognition using Combined Statistical and Geometrical Feature Vectors based on SVM

Omer Rashid, Ayoub Al-Hamadi, Axel Panning and Bernd Michaelis

***Abstract*—**It is hard to percept the interaction process with machines when visual information is not available. In this paper, we have addressed this issue to provide interaction through visual techniques. Posture recognition is done for American Sign Language to recognize static alphabets and numbers. 3D information is exploited to obtain segmentation of hands and face using normal Gaussian distribution and depth information. Features for posture recognition are computed using statistical and geometrical properties which are translation, rotation and scale invariant. Hu-Moment as statistical features and; circularity and rectangularity as geometrical features are incorporated to build the feature vectors. These feature vectors are used to train SVM for classification that recognizes static alphabets and numbers. For the alphabets, curvature analysis is carried out to reduce the misclassifications. The experimental results show that proposed system recognizes posture symbols by achieving recognition rate of 98.65% and 98.6% for ASL alphabets and numbers respectively.

***Keywords*—**Feature Extraction, Posture Recognition, Pattern Recognition, Application.

I. INTRODUCTION

Communication is a fundamental requirement for survival and interaction provides a mean to communicate. Naturally, different communication ways are used for interaction such as language, eyes, body movement, facial expression, hand gesture and posture. However, with machines, (i.e. computers) the most commonly used devices used to interact are keyboard and mouse or joystick.

Interaction with computers emerged itself as a new field of research called Human Computer Interaction (HCI) with a motivation to investigate new methodologies and techniques to improve interaction between human and computers naturally. HCI envisions to research on visual interaction with machines through virtual reality, virtual interfaces, haptic interfaces and exploiting computer vision. An intensive research has been done in the field of computer vision based on gesture and posture recognition since last decade. Many pioneering techniques have been proposed to solve the research challenges under various works and application domains. However, due to the rapid development in hardware technology (i.e. high processing and storage component), it empowers research in vision more dynamically. Thus, the challenge of finding a natural mean of interaction still remains and yet to address.

Sign language recognition is an application area for HCI to communicate with computers and for the detection of sign language symbols. Sign language is categorized into three main groups namely Finger spelling, Word level sign and Non manual features [1]. Finger spelling is used to convey the words letter by letter. The major communication is done through Word level sign vocabulary whereas Non manual features include facial expressions, mouth and body position. The techniques for posture recognition with sign languages are reviewed for finger spelling to understand the research issues. The motivation behind this research is to develop a recognition system which works more robustly with high recognition rates.

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) model is used for the recognition of Arabic Sign Language in [2]. In this approach, gloves are used for the detection of fingertip and wrist location with six different colors. However, the use of colored gloves avoids the segmentation problem and helps the system to obtain good features. Handouyahia et al. [3] presents a recognition system based on the shape description using size functions for International Sign Language (ISL). They have used Neural Network (NN) to train the alphabets because it can learn from the features computed for the sign languages. However, the drawback of the system is that the computed features are not rotation invariant. So, a change in the rotation of the hand leads to recognition of a different sign.

Other approach includes the Elliptic Fourier Descriptor (EFD) used by Malassiotis and Strintzis [4] for 3D hand posture recognition. In their system, they have used orientation and silhouettes from the hand to recognize 3D hand postures. Similarly, Licsar and Sziranyi [5] used Fourier coefficients to represent hand shape in their system which enables them to analyze hand gestures for the recognition. Freeman and Roth [6] used orientation histogram for the classification of gesture symbols, but huge training data is used to solve the orientation problem and to avoid the misclassification among symbols.

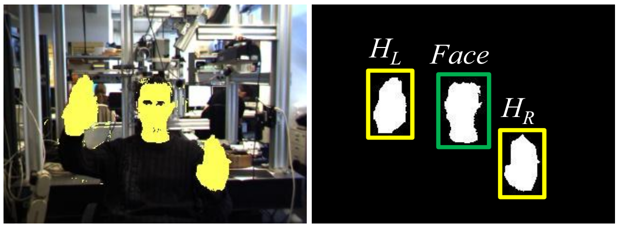
In this paper, an approach is proposed for posture recognition which includes a set of thirteen alphabets and seven numbers in American Sign Language. The system is based on the analysis of color image sequences with the support of depth information. Hands and face are segmented using normal Gaussian distribution which detects the skin pixels from 2D image sequences and the depth information is

O. Rashid is with the Institute for Electronics, Signal Processing and Communications (IESK) Otto-von-Guericke-University Magdeburg, Germany (corresponding author to provide phone: 0049-176-63152064; e-mail: omer.ahmad@ovgu.de).

A. Al-Hamadi and B. Michaelis are with the Institute for Electronics, Signal Processing and Communications (IESK) Otto-von-Guericke-University Magdeburg, Germany. (e-mail: Ayoub.Al-Hamadi@ovgu.de).

(a) (b)

Fig. 1 (a) Original Image (b) Normal Gaussian Distribution with depth Information



(a) (b)

Fig. 2 (a) Zero depth information Added (b) Detected hands and face.

used to help normal Gaussian distribution for building the region of interest. Posture features are computed from statistical and geometrical features of the hands and these features are used for the classification of posture symbols. The classification step is divided into two parts. The first part develops the classes for the set of alphabets based on the curvature analysis. It determines the peaks of hand (i.e. fingertips) for ASL which helps in the reduction of computation for the next step and to avoid the classes that are not mandatory to test for that posture symbol. In the second part, Support Vector Machines (SVM) is used on the set of classes to train and test the symbols.

This paper is organized as follows. Section II demonstrates proposed posture recognition system for American Sign Language. Experimental results are presented in section III. Section IV sums up with conclusion and future work.

II. PROPOSED APPROACH

In this section, components of proposed posture recognition systems are described in detail. Firstly, hands and face are extracted using color and depth information. Secondly, feature vectors are computed for the posture recognition system which exploits different properties of the hands. After that, fingertip is detected to reduce the misclassification. Finally, last part of the proposed approach is the recognition of ASL posture signs with SVM.

A. Image Segmentation and Object Detection

The acquisition is done by Bumblebee2 camera which gives 2D image sequences and depth image sequences [13]. The depth images are used to select the region of interest for segmentation of objects (i.e. hands and face). In our case, it lies in the range from minimum depth 30 cm to maximum depth 200 cm in the image. The depth information narrows down the search for the object of interest. YC_bC_r color space is used because luminance information is separated from chrominance in this color space. In this color space, skin color region lies in a small region of the chrominance components. Therefore, the luminance information is ignored to reduce effect of brightness variation. The skin color distribution is modeled by normal Gaussian distribution and is characterized by mean and variance. As chrominance components (i.e. C_b and C_r) are used, so the vector is 2D (i.e. $x = [C_b\ C_r]^T$). Mean and covariance computed from training data for distribution are written as:

$$\mu = \begin{bmatrix} 97.2 \\ 164.15 \end{bmatrix}, \qquad \Sigma = \begin{bmatrix} 241.57 & -115.44 \\ -115.44 & 208.66 \end{bmatrix} \tag{1}$$

Fig. 1(a) shows the original frame and Fig. 1(b) shows skin pixels computed using normal Gaussian distribution with the depth information. It can be seen that there are some pixels which are part of skin and are not classified as skin pixels due to the region of depth information. We refer to them as zero depth image pixels. In some cases, Bumblebee camera does not predict the depth value of pixel and mark its depth 0. This results in false detection of pixels as non-skin pixel instead of skin pixels. These depth values are considered as don't care in the classification of skin pixels and are added later. Fig. 2(a) shows the result after considering zero depth image pixels.

In the objects detection, binarization is the first step which converts the segmented image into two color image (e.g. black and white). Contours are extracted from this image by computing the chain code representation based on 8-neighbor connectivity of the segments for the identification of hands and face. There are three basic assumptions to represent hands and face. The first criterion presents that in the x-coordinates, left contour represent left hand, the middle contour represents face and right contour represent right hand. Second criterion is for hands and face that a face should be in middle of the screen. Therefore, it constraints the y-coordinates search in the middle for the face. The third criterion is the weight assignments for hands and face depending upon the area. The final detected objects (i.e. hands and face) are shown in Fig. 2 (b). The search area is built around the hands and face to find the objects in the next frames.

B. Feature Extraction

In the proposed approach, two types of the feature vectors are computed, statistical feature vectors and geometrical feature vectors. These are described in the following section.

1) Statistical Feature Vectors

We have used Hu-Moments [7] for the statistical feature vectors which are derived from basic moments. Moments are used to describe the properties of objects shape statistically. In image analysis, moments are considered as a binarized or grey level image with 2D density distribution functions. Area, mean, variance, covariance and skewness are the properties which are extracted from the moments. Hu [7] derived a set of seven moments which are translation, orientation and scale invariant. Hu invariants are extended by Maitra [8] to be invariant under image contrast. Later, Flusser and Suk [9] derived the moment invariant, that are invariant under general affine transformation. The equations of Hu-Moments are defined as:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4{\eta_{11}}^2 \tag{3}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{5}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})\,(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 \\ & -3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})\,(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 \; - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{6}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})\,(\eta_{21} + \eta_{03})\end{aligned} \tag{7}$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{12} - \eta_{03})\,(\eta_{30} + \eta_{12})\,[(\eta_{30} + \eta_{12})^2 \\ & -3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})\,(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2\,(\eta_{21} + \eta_{03})^2]\end{aligned} \tag{8}$$

These seven moments are derived from second and third order moments. However, zero and first order moments are not used in this process. The first six Hu-Moments are invariant to reflection [10] and seventh moment change the sign. Statistical feature vectors contain the following set:

$$F_{stat} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7)^T \tag{9}$$

where ϕ_1 is the first Hu-Moment. Similar is the notation for all other features in this set.

2) *Geometrical Feature Vectors*

Geometrical feature set contains two features: circularity and rectangularity. These features are computed to exploit the hand shape with the standard shapes like circle and rectangle. This feature set varies from symbol to symbol and is useful to recognize the alphabets and numbers. The feature set of the geometrical features are as under:

$$F_{geo} = (Cir, Rect)^T \tag{10}$$

Circularity: Circularity is the measure of the shape that how much the objects shape is closer to the circle. In the ideal case, circle gives the circularity as one. The range of circularity varies from 1 to infinity. Circularity Cir is defined as:

$$Cir = \frac{Perimeter^2}{4\pi \times Area} \tag{11}$$

where *Perimeter* is the contour of the hand and *Area* is the total number of hand pixels.

Rectangularity: Rectangularity defines the measure of the shape of the object that how much its shape is closer to the rectangle. Orientation of the object is calculated by computing the angle of all the contour points using central moments. Length *l* and width *w* is calculated by the difference of largest and smallest angle in the rotation. In ideal case, rectangularity *Rect* is 1 for rectangle and varies from 0.5 to infinity. It is calculated as:

$$Rect = \frac{Area}{l \times w} \tag{12}$$

where area is the total pixels of the hand, l is the length and w is the width.

The statistical and geometrical feature vector set are combined together to form a set of feature set. It is denoted as:

$$F_{total} = F_{stat} + F_{geo} \tag{13}$$

$$F_{total} = (\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6, \phi_7, Cir, Rect)^T \tag{14}$$

F_{total} contains the features used for posture recognition.

Normalization: The normalization is done for features to keep them in a particular range. Geometrical features vector have the range up to infinity and these features are very different from each other, so they create a scalability problem. In order to keep them in same range and to combine them with statistical feature vector, normalization is carried out and is defined as:

$$Cir_{norm} = \frac{Cir - minCir}{maxCir - minCir} \tag{15}$$

$$Rect_{norm} = \frac{Rect - minRect}{maxRect - minRect} \tag{16}$$

where *minCir* and *maxCir* are the minimum and maximum circularity of the hand from all classes respectively. Cir_{norm} is the normalized circularity. The notations are the same for rectangularity. Hu-Moments are normalized by the following equation.

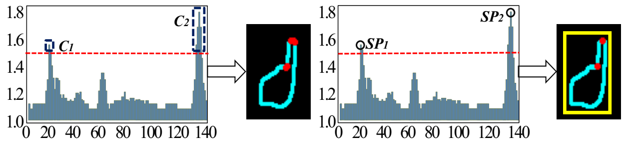
$$\phi_i = \frac{\phi_i - min\phi_{all}}{max\phi_{all} - min\phi_{all}} \tag{17}$$

where ϕ_i is the i^{th} Hu-Moment feature. $min\phi_{all}$ and $max\phi_{all}$ are the minimum and maximum values from the set of all classes respectively.

C. *Fingertip Detection*

Contour of hand is used for the detection of fingertip; where the curvature is computed for every contour point of hand. Mathematically, curvature gives the ratio of length (i.e. sum of distances that a curve has) and displacement measures the distance from the first to last point if curve covers a straight line.

Curvature is computed from the following equation:



(a) (b)

Fig. 3 (a) It shows clusters ($i.e.$ C_1 and C_2) whose threshold is above 1.5. (b) Maximum local extreme selected contour point ($i.e.$ SP_1 and SP_2) from these clusters. Red points show values above threshold 1.5 (i.e. candidates for the fingertip).

$$Curv(k) = length/displacement \tag{18}$$

$$length = \sum_{i=(k-n/2)}^{i=(k+n/2)} \|(P_i - P_{i+1})\| \tag{19}$$

$$displacement = \|(P_{(k-n/2)} - P_{(k+n/2)})\| \tag{20}$$

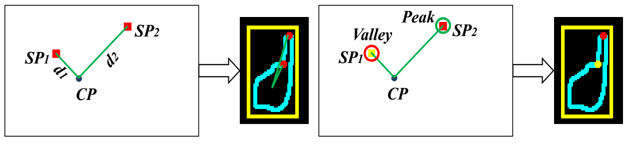
where k is the boundary point of object at which curvature is computed, $Curv(k)$ is the curvature, n is the total number of pixels used for curvature, P_i and $P_{(i+1)}$ are the objects boundary points for which x and y are object pixels. The difference between the first contour point and the last contour point is the displacement.

The objective is to find high curvature values from the contour points which results in detection of peaks from hands contour. The peaks of hand's contour represent the fingertip. Number of contour points taken for the curvature is adaptively set by exploiting the depth information. Depth information tells us about how far or close an object is from the camera and the selection of contour points is adaptively set for objects of interest ranging from 30cm to 200cm. Besides, the scaling problem is also resolved in this manner. The curvature values above 1.5 are selected as a candidate for fingertip. The experimental results show contour of the left hand with threshold greater than 1.5. In Fig. 3 (a), there are two clusters named C_1 and C_2 and the maximum value from these clusters is selected using the maximum local extreme value. The resulted points are marked as a fingertip (i.e. SP_1 and SP_2) as shown in Fig. 3(b).

The experimental results show that fingertip can be detected wrong from the algorithm because it can detect the peaks (i.e. ∩) and valleys (i.e. ∪) both as the fingertip. The next step is to remove the valleys from being detected as a fingertip. For this purpose, we took the selected contour points(i.e. SP_1 and SP_2) and compute the distance from the center point CP of the object as shown in Fig. 4(a). The normalization is done and these points are scaled ranging from 0 to 1. We pick the points whose values are greater than 0.5 for fingertip detection. In Fig. 4(b), the red mark represents the fingertip whereas the yellow mark points to a valley (not a fingertip). Thus, in this way, we are able to detect the fingertips correctly using set criteria.

D. Classification

Classification is the last step in which a posture symbol is



(a) (b)

Fig. 4 (a) Red points show the selected contour points ($i.e.$ SP_1 and SP_2). Distance is calculated from center point (CP) and normalization is done. (b) Normalized values greater than 0.5 are detected as fingertip (i.e. peak) and marked by red point. Yellow marks represent the values less than 0.5.

assigned to one of the predefined classes. A set of thirteen ASL alphabets (i.e. A, B, C, D, H, I, L, P, Q, U, V, W and Y) and seven ASL numbers (i.e. 0-6) are recognized using SVM.

As mentioned earlier, the classification phase contains two parts. Curvature is analyzed in the first part for ASL alphabets where as SVM classifier is used in the second part for both ASL alphabets and numbers. The reason for not putting these numbers with alphabets is that some numbers are very similar to alphabets and therefore, it is hard to classify these signs. For example, 'D' and '1' are same with a small change of thumb. Moreover, unlike the alphabets, ASL numbers are not categorized into groups and the classification is carried out for a single group. In this way, no curvature analysis is computed for ASL numbers and it includes only SVM classifier phase.

1) Curvature Analysis

In ASL classification phase, we have used curvature value greater than 1.5 for detecting the number of fingertips to create groups. These groups are shown in Table I. This analysis is done to reduce the number of posture signs in each group and to avoid the misclassifications. Therefore, before classifying the alphabets by SVM, four groups are made according to the numbers of fingertips detected in the hand. However, for ASL numbers, we classify them with a single classifier.

TABLE I
NUMBER OF DETECTED FINGERTIPS IN POSTURE ALPHABETS

Group Nr.	Fingers	Posture Symbols
1	0	A, B
2	1	A, B, D, H, I, U
3	2	C, L, P, Q, V,Y
4	3	W

2) SVM Classifier

Support Vector Machines (SVM) is a supervised learning technique for optimal modeling of data. It learns decision function and separates data class to the maximum width. SVM learner defines the hyper-planes for the data and maximum margin is found between these hyper-planes. Because of the maximum separation of hyper-planes, it is also considered as a margin classifier. Margin of the hyper-plane is the minimum distance between hyper-plane and the support vectors and this margin is maximized in it.

SVM is a well suited classifier where features are large in number because they are robust to the curse of dimensionality. Kernel function is the computation of the inner product $\Phi(x) \cdot \Phi(y)$ directly from the input. One of the characteristics of using the kernel is that there is no need to explicitly represent

TABLE II
CONFUSION MATRIX WITH ONE FINGERTIP DETECTED

Symbol	A	B	D	I	H/U
A	99.8	0.0	0.0	0.0	0.2
B	0.0	98.18	1.0	0.0	0.82
D	0.0	0.0	98.67	1.33	0.0
I	0.58	0.0	0.8	98.62	0.0
H/U	0.0	3.08	0.0	0.24	96.68

TABLE III
CONFUSION MATRIX WITH TWO FINGERTIPS DETECTED

Symbol	C	L	P	Q	V	Y
C	98.65	0.25	0.0	0.75	0.0	0.35
L	0.38	98.5	0.0	0.76	0.0	0.36
P	0.0	0.0	98.74	1.26	0.0	0.0
Q	0.0	0.0	3.78	96.22	0.0	0.0
V	0.20	0.0	0.0	0.0	99.35	0.45
Y	0.0	0.0	0.0	0.0	0.7	99.3

the mapped feature space.

Statistical and geometrical features are used to train and classify signs using SVM which makes a good combination for the recognition. Radial Basis Function (RBF) Gaussian kernel is used in the proposed approach which has performed robustly with given number of features (i.e. statistical and geometrical features) and provided optimum results as compared to other kernels. For the optimization, kernel width σ and the penalty parameter C are dealt with $\sigma = 3, C = 5$. More details about SVM can be found in [11], [12].

III. Experimental Results

A database is built to train posture signs which contain 3000 samples taken from eight persons on a set of thirteen ASL alphabets and seven numbers. Classification results are based on 2000 test samples from five persons and the test data used is entirely different from the training data. As the computed features set are invariant to translation, rotation and scaling, therefore posture signs are tested for these properties.

Experimental result shows the probability of posture classification for each class in the group and it is achieved for test data by the analysis of confusion matrixes. The calculated results include the test posture samples (i.e. alphabets and numbers) with rotation and scaling. The diagonal elements in the confusion matrixes represent the percentage probability of each class in the group. Misclassifications between the different classes are shown by the non-diagonal elements. Feature vector set for posture recognition contains the statistical feature vectors and geometrical feature vectors, so the computed confusion matrix from these features gives an inside view about how different posture symbols are similar to each other. Confusion matrixes and classification probabilities of the groups for ASL alphabets are described here:

Group 1 (No Fingertip Detected): ASL alphabets in this group are ‘A’ and ‘B’. There is no misclassification between these two classes and these posture symbols are very different from each other. Precisely, the geometrical feature vectors of these signs are different from each other but statistical feature vectors show correlation in Hu-Moments from (i.e. $\phi_3, \phi_4, \phi_5, \phi_6$and ϕ_7).

TABLE IV
CONFUSION MATRIX OF FINGERTIPS (i.e. ALL)

Fingertip/GroupNr.	1	2	3	4
0	99.77	0.23	0.0	0.0
1	1.26	96.8	1.94	0.0
2	0.0	0.0	95.01	4.99
3	0.0	0.0	0.9	99.1

TABLE V
CONFUSION MATRIX OF ASL NUMBERS

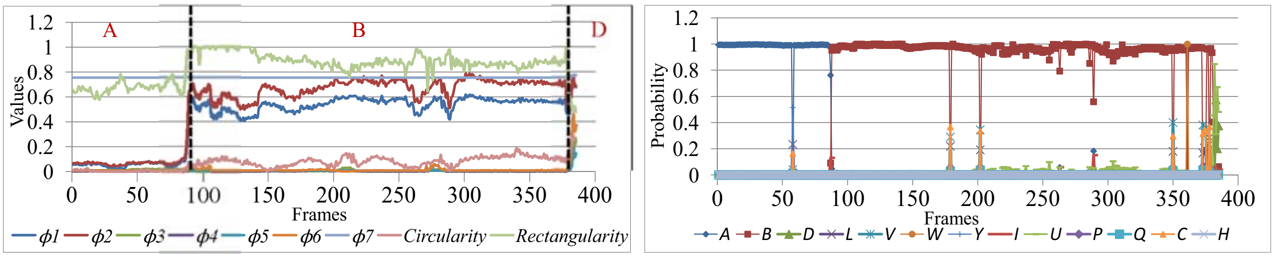
Numbers	0	1	2	3	4	5	6
0	99.8	0.2	0.0	0.0	0.0	0.0	0.0
1	0.3	99.4	0.3	0.0	0.0	0.0	0.0
2	0.0	0.0	98.3	0.4	0.0	0.0	1.3
3	0.0	0.0	0.4	98.2	0.9	0.0	0.5
4	0.0	0.0	0.0	0.2	98.2	1.6	0.0
5	0.0	0.0	0.0	0.0	2.4	97.6	0.0
6	0.0	0.0	0.8	0.6	0.0	0.0	98.6

Group 2 (One Fingertip Detected): Table II shows the confusion matrix of the classes with one fingertip detected. The result of misclassification shows the tendency of a posture symbol towards its nearby posture class. Posture symbols are tested on different orientations and back and forth movements. It can be seen that alphabet ‘A’ results in least misclassification with the other posture symbols because alphabet ‘A’ is different from other postures in this group. ‘H’/‘U’ has the maximum misclassification with the other posture alphabets. It is observed that the misclassification of ‘H’/‘U’ with ‘B’ is occurred during the back and forth movement. In general, there are few misclassifications between these posture signs because of the features which are translation, rotation and scale invariant.

Group 3 (Two Fingertips Detected): Confusion matrix of group with two detected fingertips is shown in Table III. The posture symbols in this group are tested for translation, orientation and scaling. The presented results show that the highest misclassification exists between ‘P’ and ‘Q’. It is due to the reason that these two signs are not very different in shape and geometry. Besides, statistical features in this group lie in the same range. Therefore, a strong correlation exists between the symbols in this group which leads to the misclassification between them.

Group 4 (Three Fingertips Detected): The posture symbol ‘W’ only falls in the category of three fingertips detection. Therefore, it always results in classification of alphabet ‘W’.

Misclassification of Groups: The fingertip probability classification of a group with the other groups is shown in Table IV. It is observed that the misclassification exists in the neighboring groups and is due to the reason that only +/- 1 detected fingertip is wrong or wrongly detected for posture signs. A strong misclassification between the detection of group 3 and group 4 fingertips can be seen in Table IV. Wrong fingertip detection from group 3 leads to the detection of three fingertips and thus ‘W’ is recognized from posture sign. Similarly, if two fingertips are detected in group 4, it leads to the classification from group with two fingertips detected class signs.



(a) (b)

Fig. 5: (a) Graph shows the statistical and geometrical feature vector set of the posture signs ‘A’, ‘B’ and ‘D’. (b) Classification probability of the test sequence. Blue curve shows the highest probability in the initial frames which classifies ‘A’; classification for ‘B’ sign is shown by the brown curve and ‘D’ is shown in the last frames by light green curve.

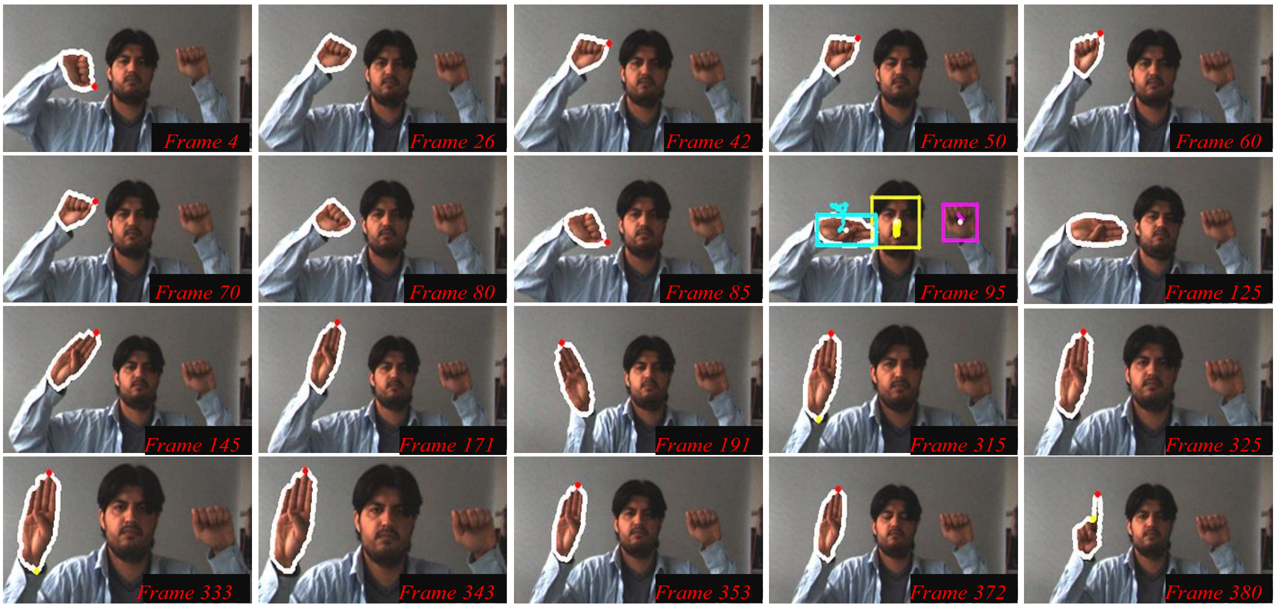
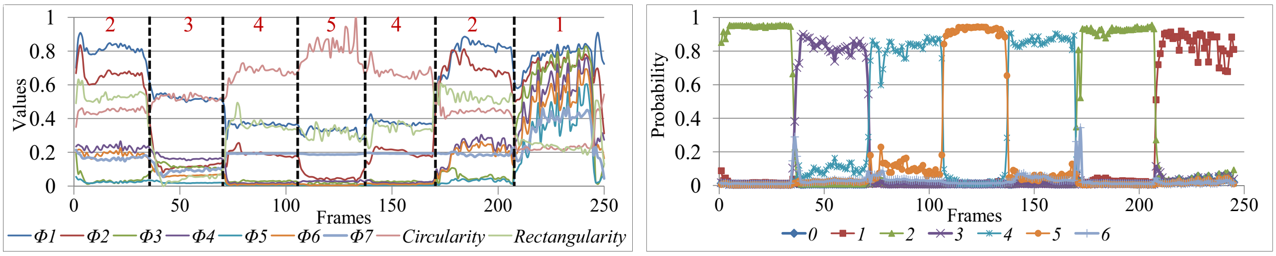


Fig. 6 Test Sequence “ABD-Sequence” for the postures signs ‘A’, ‘B’ and ‘D’ with different rotations, scaling and partial occlusion is presented.

ASL Numbers: Table V shows the confusion matrix of the classes for ASL numbers and these are tested for translation, orientation and scaling. The presented results show the least misclassification of number ‘0’ with the other classes because its geometrical features are entirely different from the other classes. Highest misclassification exists between numbers ‘4’ and ‘5’ as the similarity between these signs (i.e. thumb in number ‘5’ is open) is very high. Other misclassifications exists between the numbers ‘3’ and ‘6’.

For the posture recognition, classification results based on statistical and geometrical feature vectors are shown in Fig. 5(b) and Fig. 7(b). This result shows that the SVM clearly defines the boundaries between different classes in a group. In these figures, Y-axis shows the probability of the classes and time domain (frames) are represented in the X-axis. Probabilities computed by SVM of the resultant posture alphabets are higher due to the separation among the posture classes in respective group.

Test Sequence 1 with Classification Results: In Fig. 5(a), major part of graph includes posture signs ‘A’, ‘B’ and last frames show posture symbol ‘D’. Posture signs ‘A’ and ‘B’ are the two signs that are categorized in two groups (i.e. no fingertip detection and one fingertip detected). The symbols in this sequence are tested for rotation and, back and forth movement. Fig. 6 presents the test sequence with detected contour and fingertips of left hand. It can also be seen that the left hand and right hand can represent different posture signs but here the results of left hand is presented. However, it can be seen that features of posture signs does not affect much under rotation and scaling. Fig. 5(b) presents the classification probabilities for test sequence. The classification presents good results because the probability of resultant class with respect to other classes is high. The discrimination power of SVM can be seen from this behavior and it classifies the posture signs ‘A’ and ‘B’ correctly. In the sequence, posture sign change from ‘A’ to ‘B’ in frame 90, followed by another symbol change at frame 377 from ‘B’ to ‘D’. Posture sign ‘B’ is detected robustly despite of orientation, scaling and occlusion. However, misclassifications between the groups



(a) (b)

Fig. 7(a) Graph shows the statistical and geometrical feature vector for posture sign '1', '2', '3', '4' and '5' (b) Classification probability of the sequence is presented. The graph shows the classification of seven numbers in sequence starting from '2' and ends up at '1'. The numbers classified are '1', '2', '3', '4' ,'5' and the numbers '4' and '2' occur twice in sequence.

Fig. 8 Test Sequence "Number-Sequence" is tested under rotation and by changing a lot of numbers during sequence. The recognized number is also shown in every frame.

can be seen from the graph due to false fingertip detection and segmentation. For example, in the frames where no fingertip is detected, posture signs 'A' and 'B' are classified correctly but the misclassifications are observed with other signs in the group with one fingertip detected.

Test Sequence 2 with Classification Results: Fig. 7(a) presents a test sequence for ASL numbers. The sequence is tested under rotation. Moreover, the sequence is tested by rapidly changing signs. The sequence in Fig. 8 contains ASL numbers '1','2','3','4','5' in which multiple entries of '2' and '4' are observed. SVM classifies the numbers correctly and the recognition results are also shown in the sequence. The results in Fig. 7(a) are presented for left hand.

Fig. 7(b) shows the classification probabilities of posture number signs for test sequence in Fig. 8. The rapid changes in the signs are seen in this sequence. In the graph, a significant difference in probabilities can be observed between probabilities of classified and non-classified numbers. For example, first recognized number is '2' up to frame 35 and it has the probability greater than 80%. Similar is the case with other classified numbers. No misclassification occurs in the sequence expect when changing from one number to the next. The legend in the graph shows total number of numbers in the database.

IV. CONCLUSION AND FUTURE WORK

A framework is presented for the posture recognition which recognizes ASL alphabets and numbers. The proposed framework achieves good recognition results under rotation, and back and forth movements. Statistical and geometrical features are selected which are translation, orientation and scaling invariant. The fingertip is detected for ASL alphabets and used as a feature to reduce the misclassifications by grouping. SVM is applied for the recognition of ASL signs where the RBF kernel is used.

The extension of the future work is manifold in the field of posture recognition. It can be extended by adding rest of signs for ASL. Further, other sign languages can also be incorporated in it, therefore a multilingual posture recognition system is built. Furthermore, this framework can be extended by including the words and sentences of sign languages along with facial features. The aim behind the incorporation of multi-model system is to envision a wide range of real-time applications and addresses the realistic challenge which is the main motivation of HCI research.

ACKNOWLEDGEMENT

This work is supported by Transregional Collaborative Research Center SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Foundation (DFG).

REFERENCES

[1] R. Bowden, A. Zisserman, T. Kadir and M. Brady, "Vision Based Interpretation of Natural Sign Languages," in *Proc. 3rd International Conference on Computer Vision Systems,* 2003, pp. 391-401.

[2] M. Hussain, "Automatic Recognition of Sign Language Gestures," Master Thesis, Jordan University of Science and Technology, 1999.

[3] M. Handouyahia, D. Ziou and S. Wang, "Sign Language Recognition Using Moment-Based Size Functions," in *Proc. International Conference of Vision Interface*, 1999, pp. 210-216

[4] S. Malassiotis and M. Strintzis, “Real-time Hand Posture Recognition using Range Data,” in *Image and Vision Computing*, Vol. 26, No. 7 pp. 1027-1037, 2008.

[5] A. Licsar and T. Sziranyi, “Supervised Training Based Hand Gesture Recognition System,” in *Proc. International Conference on Pattern Recognition,* 2002, pp. 999-1002.

[6] W. Freeman and M. Roth, “Orientation histograms for hand gesture recognition,” in *Proc. International Workshop on Automatic Face and Gesture Recognition*, 1994, pp. 296-301.

[7] M. Hu. “Visual Pattern Recognition by Moment Invariants,” in *IRE Transaction on Information Theory*, Vol. 8, No. 2, pp. 179-187, 1962.

[8] S. Maitra, “Moment Invariants,” in *Proc. of the IEEE*, Vol. 67, pp. 697-699, 1979.

[9] J. Flusser and T. Suk, “Pattern Recognition by Affine Moment Invariants,” in *Journal of Pattern Recognition*, Vol. 26, No. 1, pp. 167-174, 1993.

[10] J. Davis and G. Bradski, “Real-time Motion Template Gradients using Intel CVLib,” in *Proc. of IEEE ICCV Workshop on Framerate Vision*, 1999.

[11] N. Cristianini and J. Taylor, *An Introduction to Support Vector Machines and other kernel based learning methods*, Cambridge University Press, 2001.

[12] C.J. Lin and R. Weng, “Simple Probabilistic Predictions for Support Vector Regression,” in *Technical Report*, Department of Computer Science, National Taiwan University, 2004.

[13] Point Gray Research. (2008, Nov 8). Bumblebee2 stereo vision camera. Available: www.ptgrey.com/products/Point_Grey_stereo_catalog.pdf