Sign Language Recognition System For Deaf And Dumb People

Sakshi Goyal[1], Ishita Sharma[2], Shanu Sharma[3]

[1]*Student, CSE Department, ASET,Amity University, Noida, Uttar Pradesh, India,*
[2]*Student, CSE Department, ASET,Amity University, Noida, Uttar Pradesh, India,*
[3]*Assistant Professor, CSE Department, ASET, Amity University, Noida, Uttar Pradesh, India*

Abstract

The Sign language is very important for people who have hearing and speaking deficiency generally called Deaf And Mute. It is the only mode of communication for such people to convey their messages and it becomes very important for people to understand their language. This paper proposes the method or algorithm for an application which would help in recognising the different signs which is called Indian Sign Language. The images are of the palm side of right and left hand and are loaded at runtime. The method has been developed with respect to single user. The real time images will be captured first and then stored in directory and on recently captured image and feature extraction will take place to identify which sign has been articulated by the user through SIFT (scale invariance Fourier transform) algorithm. The comparisons will be performed in arrears and then after comparison the result will be produced in accordance through matched keypoints from the input image to the image stored for a specific letter already in the directory or the database the outputs for the following can be seen in below sections. There are 26 signs in Indian Sign Language corresponding to each alphabet out which the proposed algorithm provided with 95% accurate results for 9 alphabets with their images captured at every possible angle and distance i.e. for every alphabet even if have approximately 5 images at different angles and distances then the algorithm is working accurately for 45 types of inputs.

Keywords – Indian Sign Language, Feature Extraction, Keypoint Matching, Sign/Gesture Recognition

1. Introduction

To establish a communication or interaction with Deaf and Mute people is of utter importance nowadays. These people interact through hand gestures or signs. Gestures are basically the physical action form performed by a person to convey some meaningful information. Gestures are a powerful means of communication among humans. In fact gesturing is so deeply rooted in our communication that people often continue gesturing when speaking on the telephone. There are various signs which express complex meanings and recognising them is a challenging task for people who have no understanding for that language.

It becomes difficult finding a well experienced and educated translator for the sign language every time and everywhere but human-computer interaction system for this can be installed anywhere possible. The motivation for developing such helpful application came from the fact that it would prove to be of utmost importance for socially aiding people and how it would help increasingly for social awareness as well. The remarkable ability of the human vision is the gesture recognition, it is noticeable mainly in deaf people when they communicating with each other via sign language and with hearing people as well. In this paper we take up one of the social challenges to give this set of mass a permanent solution in communicating with normal human beings.

Sign language is categorized in accordance to regions like Indian, American, Chinese, Arabic and so on and researches on hand gesture recognition, pattern recognitions, image processing have been carried by supposedly countries as well to improve the applications and bring them to the best levels.

2. Literature Survey

As mentioned in Introduction that numbers of researches have been carried out as it has become a very influential topic and has been gaining heights of increasing interest. Some methods are explained below:

The paper Real Time Hand Gesture Recognition Paper included the algorithm in which first the video was captured and then divided into various frames and the frame with the image was extracted and further from that frame various features like Difference of Guassian. Scale space Feature Detector and etc were extracted though SIFT which helped in gesture recognition[1].

A different method had been developed by Archana S Ghotkar, Rucha Khatal, Sanjana Khupase, Surbhi Asati and MIthila Hadop through Hand Gesture Recognition for Indian Sign Language consisted of use of Camshift and HSV

model and then recognizing gesture through Genetic Algorithm, in the following applying camshift and HSV model was difficult because making it compatible with different MATLAB versions was not easy and genetic algorithm takes huge amount of time for its development.[2]

A method had been developed by P Subha Rajan and Dr G Balakrishnan for recognising gestures for Indian Sign Language where the proposed that each gesture would be recognised through 7 bit orientation and generation process through RIGHT and LEFT scan. The following process required approximately six modules and was a tedious method of recognising signs[3].

A method had been developed by T. Shanableh for recognizing isolated Arabic sign language gestures in a user independent mode. In this method the signers wore gloves to simplify the process of segmenting out the hands of the signer via color segmentation. The effectiveness of the proposed user-independent feature extraction scheme was assessed by two different classification techniques; namely, K-NN and polynomial networks. Many researchers utilized special devices to recognize the Sign Language[4].

Byung - woo min et al, presented the visual recognition of static gesture or dynamic gesture, in which recognized hand gestures obtained from the visual images on a 2D image plane, without any external devices. Gestures were spotted by a task specific state transition based on natural human articulation[8].

Static gestures were recognized using image moments of hand posture, while dynamic gestures were recognized by analysing their moving trajectories on the Hidden Markov Models (HMMs).

3. Proposed Methodology

The proposed algorithm consisted of four major steps which are namely Image Acquisition, Feature Extraction, Orientation Detection and Gesture Recognition which is also shown in the below given Fig 1.

All of the following steps are explained in details in the later part of the paper with all the information on how the module is working and what behaviour the module is supposedly expected to portray. While deciding on the following algorithm it was observed that pre-processing steps that are to be applied on the images for removal of noise in the background was not at all required and the approach was concluded to be simple and easy to implement. The steps of the methodology are further explained in details:

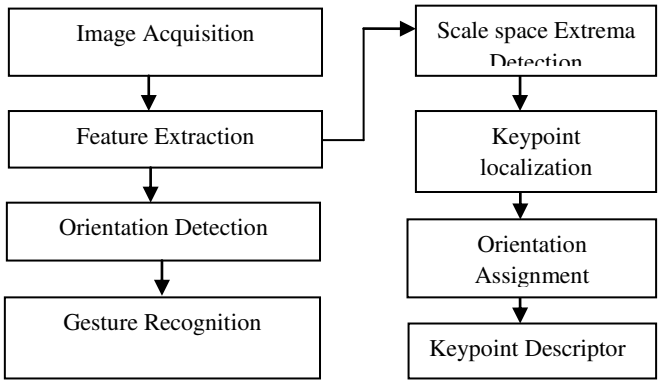

Figure 1. Flow chart of Proposed Sign Language Recognition System

3.1. Image Acquisition

The first step of Image Acquisition as the name suggests is of acquiring the image during runtime through integrated webcam and while acquiring. The images will be stored in the directory as soon as they are captured and the recently captured image will be acquired and will be compared with the images stored for specific letter in the database using the SIFT algorithm and the comparison will give the gesture that was done and the translated text for the following gesture. The images will be captured through basic code of opening a webcam through MATLAB and then capturing the image through frames per second which will be stored in another directory where all the inputs images are stored in another directory and the recent captured image is picked up and the comparison with given set of images are made.

The interface of the application is provided with the button START when the user clicks on the button it works up to open up the integrated webcam and the button changes its status to STOP and when the user is ready with the gesture it can click on the button and that frame is captured and stored in the directory.

Gesture of letters shown in fig 2. are used for testing the recognition algorithm[2].

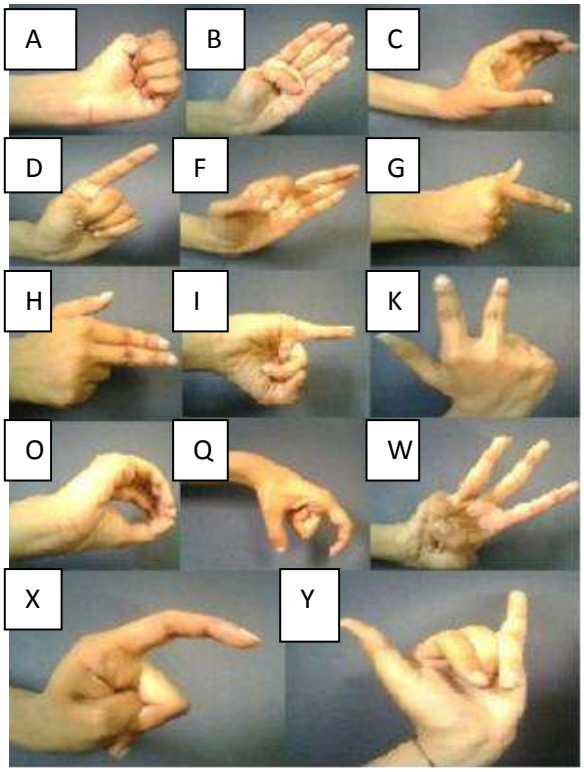

Figure 2. Gestures for different Signs

3.2. Feature Extraction

For any object there are many features, interesting points on the object, that can be extracted to provide a "feature" description of the object. SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation. The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors". Each of these feature vectors is invariant to any scaling, rotation or translation of the image. To aid the extraction of these features the SIFT algorithm applies a 4 stage filtering approach[1][5]:

3.2.1. Scale-Space Extrema Detection. This stage of the filtering attempts to identify those locations and scales that are identifiable from different views of the same object. This can be efficiently achieved using a "scale space" function. It is based on the Gaussian function. The scale space is defined by the equation 1.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (1)$$

Where * is the convolution operator, $G(x, y, \sigma)$ is a variable-scale Gaussian and $I(x, y)$ is the input image.

One can also detect the stable keypoints in an image by locating scale-space extrema, $D(x, y, \sigma)$ by computing the difference between two images, one with scale k times the other. $D(x, y, \sigma)$ is then given by equation 2:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \qquad (2)$$

To detect the local maxima and minima of $D(x, y, \sigma)$ each point is compared with its 8 neighbours at the same scale, and its 9 neighbours up and down one scale. If this value is the minimum or maximum of all these points then this point is an extrema.

3.2.2. Keypoint Localisation. This stage attempts to eliminate more points from the list of keypoints by finding those that have low contrast or are poorly localised on an edge. This is achieved by calculating the Laplacian. The location of extremum, **z**, is given by:

$$z = \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}$$

If the function value at **z** is below a threshold value then this point is excluded. This removes extrema with low contrast. To eliminate extrema based on poor localisation it is noted that in these cases there is a large principle curvature across the edge but a small curvature in the perpendicular direction in the difference of Gaussian function. If this difference is below the ratio of largest to smallest eigenvector, from the 2x2 Hessian matrix at the location and scale of the keypoint, the keypoint is rejected.

3.2.3. Orientation Assignment. This step aims to assign a consistent orientation to the keypoints based on local image properties. The keypoint descriptor, can then be represented relative to this orientation, achieving invariance to rotation. The approach taken to find an orientation is:

- Use the keypoints scale to select the Gaussian smoothed image L
- Compute gradient magnitude, *m*

$$m(x.y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

- Compute orientation, θ

$$\theta(x,y) = \tan((L(x,y+1) - L(x.y-1))/(L(x+1,y) - L(x-1,y)))$$

- Form an orientation histogram from gradient orientations of sample points and then Locate the highest peak in the histogram. Use this peak and any other local peak within 80% of the height of this peak to create a keypoint with that orientation Some points will be assigned multiple orientations Fit a parabola to the 3 histogram values closest to each peak to interpolate the peaks position.

3.2.4. Keypoint Descriptor. The local gradient data, used above, is also used to create keypoint descriptors. The gradient information is rotated to line up with the orientation of the keypoint and then weighted by a Gaussian with variance of 1.5 * keypoint scale. This data is then used to create a set of histograms over a window centred on the keypoint[6].
Keypoint descriptors typically uses a set of 16 histograms, aligned in a 4x4 grid, each with 8 orientation bins, one for each of the main compass directions and one for each of the mid-points of these directions. This results in a feature vector containing 128 elements.

These resulting vectors are known as SIFT keys and are used in a nearest-neighbours approach to identify possible objects in an image. Collections of keys that agree on a possible model are identified, when 3 or more keys agree on the model parameters this model is evident in the image with high probability. Due to the large number of SIFT keys in an image of an object, typically a 500x500 pixel image will generate in the region of 2000 features, substantial levels of occlusion are possible while the image is still recognised by this technique.

In the database we have already provided one image each for a sign to make the comparisons. After the input image is provided to the application through SIFT defined functions the application will first calculate the keypoints of the input image after the keypoints of the input image is calculated then the comparison will start. The application picks up all the images specified in database one by one find the keypoints of each image one by one and finds the number of matched keypoints , the comparisons with highest matched keypoints in an image will take the lead and will be produces as an output. The following process is explained with an example along with figures.

Supposedly if we provide the input image as the Sign/Gesture for character ‘C’

Figure 3. Keypoints calculated for input image

66 keypoints were found for the input gesture through SIFT algorithm which can be seen in fig 3. After this it picks up the first image in the database calculates the keypoints for the first image as 71 keypoints but out of the following none matched between two images. The following has been shown in the below fig 4 as well.

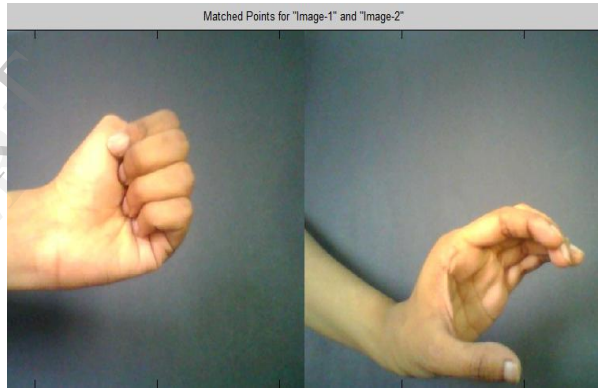

Figure 4. No keypoints matching between database and input image

Similarly it moves to the next image in database to calculate its 57 keypoints with only two matched points between the two images which is shown in fig 5.

Figure 5. Only two keypoints matching between input and database image

In fig 6 it calculates 50 keypoints with only one matched keypoint with the next image in database.

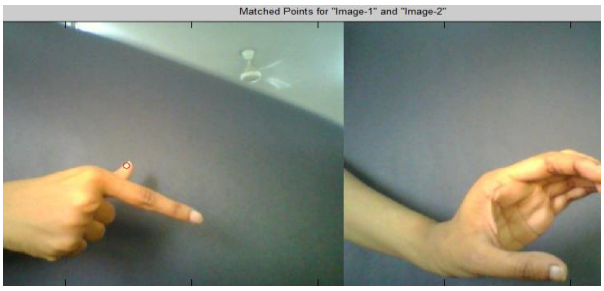

Figure 6. Only one keypoint matching between the images

And finally after calculating for every image in database it finally finds the highest keypoints matching image i.e. 8 keypoints matching it concludes the sign comparison with matched image from the database. See fig 7.

Figure 7: The highest keypoints matching image i.e. 8 keypoints is found correct and displayed

3.3. Orientation Detection

In orientation detection we will take the input of hand movement in any form or any orientation the gesture will be detected through the described section of feature extraction as the SIFT algorithm also includes the orientation assignment procedure[7].

3.4. Gesture Recognition

Finally when the whole process is complete the application will then convert the gesture into its recognized character or alphabet which might be helpful to be understood in layman's language. The following process includes passing out the single dimensional array of 26 character corresponding to alphabets has been passed where the image number stored in database is provided in the array. Supposedly if the image 2.jpg is of 'B' character in the database then 2 is passed in the array. Thus the image is picked up from the array and corresponding alphabet is displayed in the interface as shown in the given interface below.

Figure 8. The output in the interface for character 'B'

4. Conclusion

The Results shows in Fig 8 presents the output for the sign/gesture for character 'B' and Fig 3 shows the SIFT points calculated for any input image and from Figure 4 to Fig 5 is showing us the images of the comparisons happening in arrears and the algorithm is working. With our algorithm we were able to decode a video successfully with frames. The frame extraction comes within the second where the user presses the button STOP. The features were efficiently extracted using SIFT. The SIFT features described in our implementation are computed at the edges and they are invariant to image scaling, rotation.

5. Acknowledgements

We would like to thank our guide Ms. Shanu Sharma for their guidance and feedback during the course of the project. We would also like to thank our department for giving us the resources and the freedom to pursue this project.

References

[1] Pallavi Gurjal, Kiran Kunnur, "Real Time Hand Gesture Recognition using SIFT", *International Journal for Electronics and Engineering,* 2012,pp 19-33.

[2] Ghotkar, Archana S., "Hand Gesture Recognition for Indian Sign Language", *International Conference on Computer Communication and Informatics(ICCCI),* 2012, pp 1-4.

[3] Rajam, P. Subha and Dr G Balakrishnan , "Real Time Indian Sign Language Recognition System to aid Deaf and Dumb people", *13th International Conference*

on Communication Technology(ICCT), 2011,pp 737-742.

[4] Shanableh, Tamer ,T. Khaled, "Arabic sign language recognition in user independent mode", *IEEE International Conference on Intelligent and Advanced Systems,* 2007, pp 597-600.

[5] Aleem Khalid Alvi, M. Yousuf Bin Azhar, Mehmood Usman, Suleman Mumtaz, Sameer Rafiq, Razi Ur Rehman, Israr Ahmed, "Pakistan Sign Language Recognition Using Statistical Template Matching", *International Journal of Information Technology,* 2004, pp 1-12.

[6] David G.Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, 2004, pp 1-28.

[7] Yves Dufournaud,Cordelia Schmid, Radu Horaud, "Matching Images with Different Resolutions", *International Conference on Computer Vision & Pattern Recognition (CVPR '00),* 2000, pp 612-618.

[8] Byung-woo min, Ho-sub yoon, Jung soh, Takeshi ohashi and Toshiaki jima," Visual Recognition of Static/Dynamic Gesture: Gesture-Driven Editing System", *Journal of Visual Languages & Computing Volume10,Issue3,* June 1999, pp 291-309.