ISSN 2395-1621

Recognition of Hand Gesture using SIFT

Ms. Naziya R. Mulani, Mr.Vishnu C. Khade, Ms. Pratima N. Mahamuni, Mr.Vijay T.Barkade

Assistant Professor, Department of Electronics and Telecommunication,

Arvind Gavali College of Engineering, Panmalewadi, Satara, Maharashtra, India.

ABSTRACT

Infrared remote controllers have been widely used in most of electronics appliances. Most of users have difficulties of finding the appropriate controller among the increasing number of such controllers. An integrated and more convenient means has been devised to replace the existing Remote Control System (RCS). Voice or gesture recognition has been recently adapted to implement the universal RCS. Voice recognition however, may fail to provide robust performance because of the speaker variation and surrounding noise. RCSs using gloves or markers are based on the gesture recognition. They may provide accurate control performance, but the use of the gloves or markers makes the users uncomfortable. More natural way to use human actions is required for commercially worthy RCS.

Index Terms- Computer vision, hand gesture, open finger counting and matching, machine interaction.

ARTICLE INFO

Article History

Received: 25th March 2017

Received in revised form :

25th March 2017

Accepted: 25th March 2017

Published online :

4th May 2017

I. INTRODUCTION

Medical In recent year applications like Human-computer interaction and robot vision have been active research areas. The conventional HCI devices such as mouse, joysticks, roller-balls, keyboard, touch screens and electric pens has lot of limitation. As compared to conventional HCI devices, the technique using human body motions, speech and eyeball movements provide more powerful and intuitive HCI interface. [4]

An important role in human communication is played by the body language, aiming to emphasize different parts of the speech. This fact inspired many people to consider body language recognition as a mean of interacting with computers. HCI find a lot of applications in remote control as suggested. There are different types of information that HCI relay on, such as facial expression, hand posture and trajectory, eye or head movement tracking.

Human hands are frequently used for communication in daily lives. Hand gestures are the most natural easy way of communication for many deaf with sign language. Many machines are designed to be operated by gestures through mechanical and electronic interfaces. Human gestures constitute a space of motion expressed by the body, face, and/or hands. Among a variety of gestures, hand gesture is the most expressive and the most frequently used. Gestures have been used as an alternative form to communicate with computers in an easy way.

This paper is structured as follows, section-II includes comparison of techniques used for detecting hand gesture, section- III includes architecture for detecting hand gesture, implementation of Scale Invariant Feature Transform (SIFT) technique is given in section-IV, results of methodology are declared in section-V and in last section conclusion is given.

II. COMPARISON OF TECHNIQUES

TABLE I
COMPARISON OF TECHNIQUES

Sr. No.	Technique used	Additional Marker required	Segmentation Type	Gesture Size	Limitation	Accuracy
A	**Feature extraction, statistics and models**					
1	Template Based [9,10]	Glove & Vision Based	Threshold	Small	Does not work well on large posture sets due to overlapping templates	96 %
2	Feature Extraction & Analysis	Glove Based	Layered Architecture	Complex	Can be Computationally expensive	Not R

	[11]				depending on how many features are extracted	epor ted
3	Adaptive Shape Model [12]	Vision Based	Applies conture on image	Small	Tracks only the open hand and not been applied to stereo data from multiple cameras	Not Reported
4	Principal Component Analysis [14,15]	Glove & Vision Based	Computing eigen values and eigen vectors	Large	Requires normalization to keep image consistent	99 %
5	Linear Fingertip Model [16,17]	Vision Based	Uses only the fingertips as input data and permits a model that represents each finger trajectory.	Small	System did not run in real time and does not take curvilinear fingertip motion into account.	90 %
6	Causal Analysis [18]	Vision Based	Extract information from a video stream by using high-level knowledge about actions in the scene and how they relate to one another and the physical environment.	NA	Does not use hand orientation and position data or finger data & does not run in real time.	Not Reported
B	**Learning Algorithm**					
1	Neural Network [19]	Glove & Vision Based	Threshold	Small	Requires retaining of the entire network if hand postures or gestures are added or removed.	96 %
2	Hidden Markov Model [20]	Glove & Vision Based	Gmm&YCbCr Space.	Small	As with multilevel neural networks, the hidden nature of HMMs makes it difficult to observe their internal behaviour.	90 %
3	Instance Based Learning [21]	Glove Based	Include technique that represents instances as points in Euclidean Space, such as the K-Nearest Neighbour algorithm and techniques in which instances have more symbolic representation, such as case-based reasoning.	Large	Requires a large amount of primary memory as training set increases & response time issues may arise due to large amount of computation at instance classification time.	80 %
C	**Miscellaneous Techniques**					
1	Linguistic Approach [22]	Glove & Vision Based	Uses a formal grammar to represent the hand posture and gesture set.	Small	Limited to only simple hand postures.	50 %
2	Appearance Based Motion Analysis [16,23]	Vision Based	Exploits the observation that humans can recognize actions from extremely low resolution images and with little or no information about the 3D structure of the scene.	NA	Very difficult to detect small details in finger movement.	76 %
3	Spatio Temporal Vector Analysis [24]	Vision Based	Track the boundaries of the moving hand in a vision- based system.	Medium	No recognition accuracy results reported	Not Reported

With reference to above table, from point A among method 1,3,4,5 & 6 fourth method is best one with 99% accuracy but its drawback is that it requires normalization to keep the image consistent. From point B, among methods 1 & 2 first one is good with 96% accuracy but its limitation is that it requires the retraining of entire network if hand gestures are added or removed. From point C, among methods 1,2& 3 second one is better with 76% accuracy but its drawback is that it gets difficult to detect small details in finger movement. Due to this reason the scale invariant transform is used to recognize hand gesture.

III. ARCHITECTURE FOR DETECTING HAND GESTURE

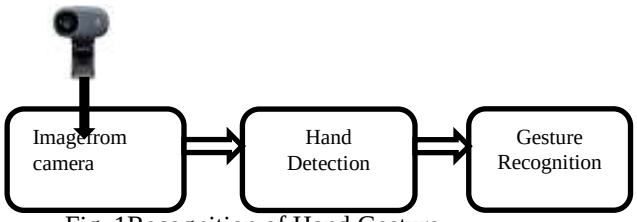

Fig. 1Recognition of Hand Gesture

Firstly the data base of hand images is created. The web camera is placed so as to take the images from a continuous video. The images captured from camera are stored, then the background is subtracted which would be easy just to take the region of hand. Finally the gesture is recognized by SIFT feature detector. For detecting the gesture following architecture is used.

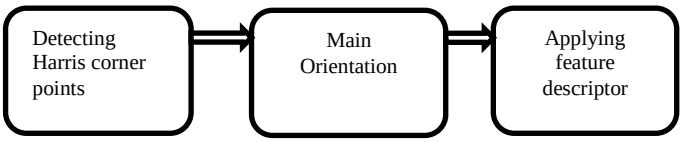

Fig. 2 Gesture Recognition

IV. PRINCIPAL FOR GESTURE RECOGNITION

4.1 Scale Invariant Feature Transform

SIFT is one of the method for detecting local feature detectors. A local feature is an image pattern which differs from its immediate neighbourhood. It is usually associated with a change of an image property or several properties simultaneously, although it is not necessarily localized exactly on this change. The image properties commonly considered are intensity, colour, and texture. Local features can be points, but also edges or small image patches. Typically, some measurements are taken from a

region centred on a local feature and converted into descriptors. The descriptors can then be used for various applications. For implementation of SIFT first the Harris corner points are to be detected and applying the Laplace transform to those points. The gradient magnitude and orientation are computed in every pixel location belonging to a selected neighbourhood of a detected local feature. Finally the descriptor is built from these detected local points. [2][1]

4.1.1Harris Detector:

The basic assumption of Harris detector is that at a corner, the intensity of an image will change in multiple directions. The detector is based on the second moment matrix M that describes the intensity change in the local neighbourhood of a point x=(x;y)

$$\mathbf{M} = g(\sigma) \begin{bmatrix} I_x^2(\mathbf{x}) & I_x I_y(\mathbf{x}) \\ I_x I_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{bmatrix}$$

------ (1)

$$I_x(\mathbf{x}) = \frac{\partial}{\partial x} I(\mathbf{x})$$

------ (2)

$$I_y(\mathbf{x}) = \frac{\partial}{\partial y} I(\mathbf{x})$$

------ (3)

$$g(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

------ (4)

First, derivatives Ix(x) and Iy(x) are computed in x and y direction. Then, I_x^2; IxIy and I_y^2 are calculated. Finally the derivatives are smoothed using Gaussian window. It has several properties that make finding corners efficient. The corners can be found in an image where the signal change is significant in both direction, i.e. the points where both eigenvalues are large.Locations of corners are quite simple to calculate after the second-moment matrix for a point has been composed. First, the 2 * 2 second-moment matrix M can be written as:

$$\mathbf{M} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}.$$

----- (5)

Harris proposed the cornerness measure R that describes the cornerness of the local neighbourhood of a point. R is computed using the trace and determinant of the matrix M. In the following equations the calculation of cornerness based on the second moment matrix is shown. The trace and the determinant of such matrix are straightforward to calculate:

$$tr(\mathbf{M}) = \lambda_1 + \lambda_2 = A + B$$

------ (6)

$$det(\mathbf{M}) = \lambda_1 \lambda_2 = AB - C^2.$$

------ (7)

Cornerness can then be easily derived. Actually there is no need to calculate the eigenvalues λ1 and λ2 since:

$$R = det(\mathbf{M}) - \alpha trace^2(\mathbf{M})$$

------- (8)

$$= \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

------- (9)

$$= AB - C^2 - \alpha(A + B)^2.$$

------- (10)

A constant α is used for balancing the terms in the equation. Typical value for α is 0:04. [6]

4.1.2 Harris Laplace Detector

With Harris corner detector, scale changes make detecting more difficult. If scale change is known, detected corners can be scaled, but if the scale change is unknown, the only option is to use multi-scale approach, where corners are detected in multiple scales. This approach also has a disadvantage: it is not feasible to have ten times more features because of the scale adaptation.

- **Automatic scale selection**

Scale invariant local features can be obtained by searching stable features in many scales, by building a scale-space presentation of an image. A scale-space presentation contains a family of smoothed images with many different scales, ϭ. Different resolution levels L are given by convolutions with the Gaussian kernel:

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

------- (11)

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

-------- (12)

where G(x; y; ϭ) is the Gaussian kernel in the scale ϭ and I represents the original image. While the original scale being ϭ0, scales for various levels are obtained by:

$$\sigma_n = k^n \sigma_0$$

-------- (13)

where n is a scale level and k is the factor of scale change between two successive scales. In the original version of Harris-Laplace detector, values for the initial scale ϭ0 and the factor k between the scales. The scale-space presentation can be used to find the characteristic scale for a local feature. The characteristic scale is found in local maxima of F(x; ϭn) where F is the normalized Laplacian:

$$| LoG(\mathbf{x}, \sigma_n) | = {\sigma_n}^2 | L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n) |.$$

-------- (14)

The same automatic scale-selection method is used for both Hessian- and Harris-based scale-invariant and affine-invariant detectors. [7]

4.1.3Main Orientation:

The scale of the keypoint is used to select the Gaussian smoothed image, L, with the closest scale, so that all computations are performed in a scale invariant manner. For each image sample, L(x, y), at this scale, the gradient magnitude, m(x, y) and orientation, Θ(x, y), is pre-computed using pixel differences:

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

----- (15)

$$\theta(x,y)=tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right).$$

------- (16)

An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint. The orientation histogram has 36 bins covering the 360 degree range of orientations. Each sample added to the histogram is weighted by its gradient magnitude and Gaussian-weighted circular window with a σ that is 1.5 times that of the scale of the keypoint. Peaks in the orientation histogram correspond to dominant directions of local gradients. The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation. Only about 15% of points are assigned multiple orientations, but these contribute significantly to the stability of matching.

4.1.4Descriptor Representation:

Figure 3 illustrates the computation of the keypoint descriptor. First the image gradient magnitude and orientations are sampled around the keypoint location, using the scale of the keypoint to select the level of Gaussian blur for the image. In order to achieve orientation in variance, the coordinates of the descriptor and the gradient orientation are rotated relative to the keypoint orientation. These are illustrated with small arrows at each sample location on the left side of figure 3.

A Gaussian weighting function with σ equal to one half the width of the descriptor window is used to assign a weight to the magnitude of each sample point. This is illustrated with a circular window on the left side of figure 3, although, of course, the weight falls of smoothly. The purpose of this Gaussian window is to avoid sudden changes in the descriptor with small changes in the position of the window, and to give less emphasis to gradients that are from the centre of the descriptor.

The keypoint descriptor is shown on the right side of figure 3. It allows for significant shift in gradient position by creating orientation histogram over 4x4 sample regions. The figure shows eight directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of that histogram entry. A gradient sample on the left can shift up to 4 sample positions while still contributing to the same histogram on the right, thereby achieving the objective of allowing for larger local positional shifts.

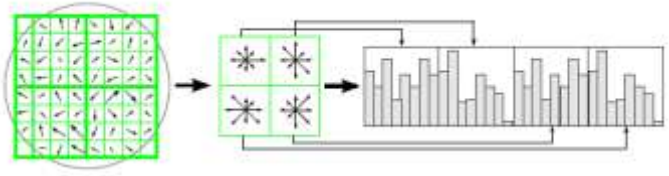

Fig. 3 Representation of Descriptor

V. RESULTS

Here the images of hand from all closed fingers to all open fingers are used to detect the gesture. The images are for both i.e for left and right hands.

Features for Left Hand:

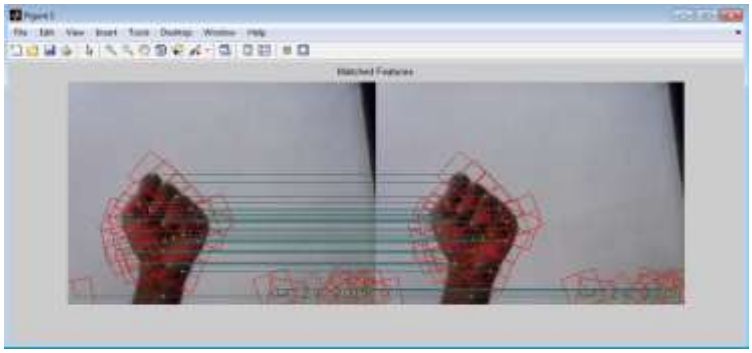

FIG. 4 Matched Features for all Fingers Closed

Fig. 5 Matched Features for all Two Fingers Raised

Fig. 6 Matched Features for Three Fingers Raised

Features for Right Hand

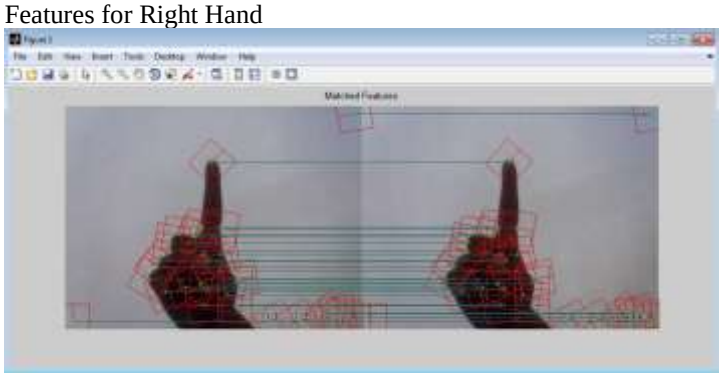

Fig. 7 Matched Features for One Fingers Raised

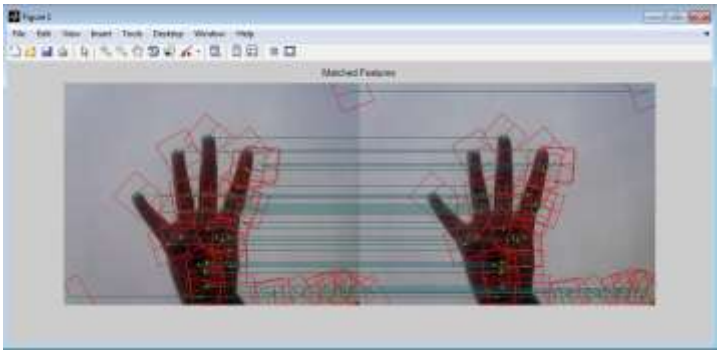

Fig. 8 Matched Features for Four Fingers Raised

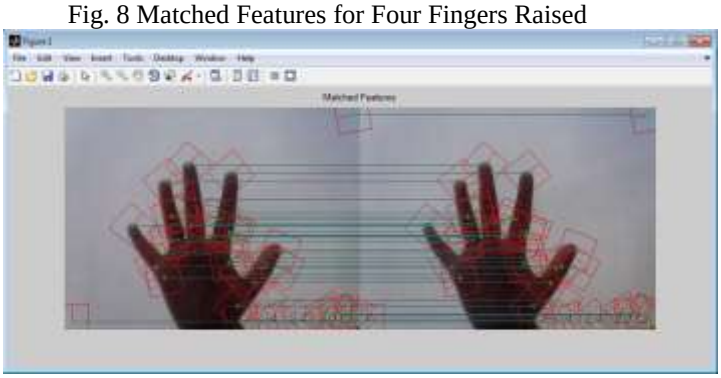

Fig. 9 Matched Features for Five Fingers Raised

RESULTS of Matched Features using GUI:

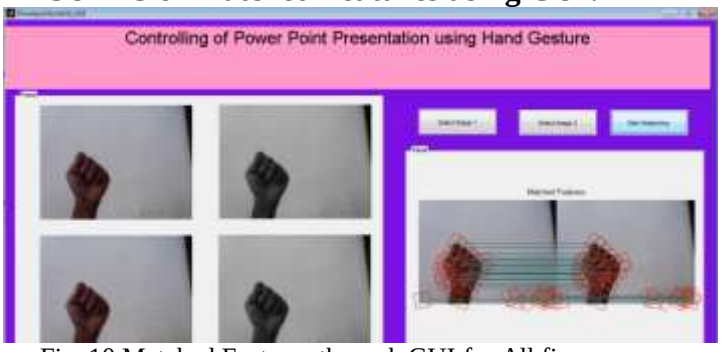

Fig. 10 Matched Features through GUI for All fingers closed

Fig. 11 Matched Features through GUI for All Fingers raised

IV. CONCLUSION

This paper gives information regarding Scale Invariant Feature Transform, through which the hand gesture can be detected very easily. This technique also provides the matching between two images. This technique gives near about 93% accuracy as compared with PCA technique which gives 99% accuracy but its main drawback is to always normalize the image. This system is implemented in GUI.

REFERENCES

[1] Professor Joni Kämäräinen, 'Comparison of local feature detectors and descriptors for visual object categorization.' 2011.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Nasser H. Dardas and Nicolas D. Georganas," Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques", IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 60, NO. 11, NOVEMBER 2011. Page no. 3592

[4] Joseph J. La Vialo Jr. 'A Survey of Hand Posture and Gesture Recognition Technique and Technology' June 1999.

[5] Noor Adnan Ibraheem and RafiqulZaman Khan ' **Survey on Various Gesture Recognition Technologies and Techniques**' *International Journal of Computer Applications (0975 – 8887) Volume 50 – No.7, July 2012*

[6] C. Harris and M. Stephens. A combined corner and edge detector. In Alvey Vision Conference, pages 147–151, 1988.

[7] T. Lindeberg. Feature detection and ridge detection with automatic scale selection. International Journal of Computer Vision, 30(2):79–116, 1998.

[8] PrateemChakraborty, PrashantSarawgi, Ankit Mehrotra, Gaurav Agarwal, Ratika Pradhan 'Hand Gesture Recognition: A Comparative Study' Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol IIMECS 2008, 19-21 March, 2008, Hong Kong.

[9] Sturman, David J. Whole-hand Input. Ph.D dissertation, Massachusetts Institute of Technology, 1992.

[10] Watson, Richard. A Survey of Gesture Recognition Techniques. Technical Report TCD-CS-93-11, Department of Computer Science, Trinity College Dublin, 1993.

[11] Rubine, Dean. Specifing Gestures by Example.In Proceedings of SIGGRAPH' 91, ACM Press, 329-337, 1991.

[12] Cootes, T.F., and C.J. Taylor. Active Shape Models – 'Smart Snakes'.In Proceedings of the British Machine Vision Conference, Springer-Verlag, 266-275, 1992.

[13] Cootes, T.F., and C.J. Taylor. Active Shape Models – 'Smart Snakes'.In Proceedings of the British Machine Vision Conference, Springer-Verlag, 266-275, 1992.

[14] Joliffe, I. T. Principal Component Analysis. Springer-Verlag, New York, 1986.

[15] Birk, Henrik, Thomas B. Moeslund, and Claus B. Madsen. Real-Time Recognition of Hand Alphabet Gestures Using Principal Component Analysis. In Proceedings of The 10th Scandinavian Conference on Image Analysis, 1997.

[16] Davis, James, and Mubarak Shah. Gesture Recognition.Technical Report, Department of

Computer Science, University of Central Florida, CS-TR-93-11, 1993.

[17] Glassner, Andrew. Principles of Digital Image Synthesis. Morgan Kaufman, San Francisco, 1995

[18] Brand, Matthew, Lawrence Birnbaum, and Paul Cooper. Sensible Scenes: Visual Understanding of Complex Structures Through Causal Analysis. In Proceedings of the 1993 AAAI Conference, 49-56,

[19] Russell, Stuart, and Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, Englewood Cliffs, NJ,1995.

[20] Charniak, Eugene. Statistical Language Learning. MIT Press, Cambridge, 1993.

[21] Mitchell, Tom M. Machine Learning. McGraw-Hill, Boston, 1997.

[22] Hand, Chris, Ian Sexton, and Michael Mullan. A Linguistic Approach to the Recognition of Hand Gestures.In Proceedings of the Designing Future Interaction Conference, University of Warwick, UK, 1994.

