

Virtual Interactive Hand Gestures Recognition System in Real Time Environment

T. B. Patil¹, Aakash Jain¹, Supriya C. Sawant², Debnath Bhattacharyya³ and Hye-Jin Kim⁴

¹Department of Information Technology, ²Department of Electronics and Tele-Communication, ³Department of Computer Science and Engineering, ⁴Business Administration Research Institute

Bharati Vidyapeeth Deemed University College of Engineering,
Pune-411043, India, SKNCOE, Vadgaon (Bk), Off Sinhgad Road,
Pune-411041, India, Vignan's Institute of Information Technology,
Visakhapatnam-530049, AP, India, Sungshin W. University,
2, Bomun-ro 34da-gil, Seongbuk-gu, Seoul
tbpatil@bvucoep.edu.in, aakash.jain0901@gmail.com,
supriya.sawant605@gmail.com, debnathb@gmail.com, hyejinaa@daum.net
(Corresponding Author)

Abstract

Normal human Beings can communicate with each other with the help of various different languages, however, the people who can't speak have different sign languages to communicate with others. But a major setback of sign language is that only people those who know sign language can communicate with them. Individuals habitually use gestures to interconnect. Gestures are used for directing to an individual, to get his devotion and convey statistics about temporal and spatial physiognomies. Gesticulating does not only elaborate verbal language, but it is part of the language production practice. In our paper we recommend a scheme which will be able to convert the sign language into words or sentences. The system takes inputs several hand gestures then processes its meaning and verifies it with the stored gestures in database and results the corresponding output on the screen.

Keywords: Database, Comparison, OpenCV, Image Processing, Convex Hull, Hand Gestures, Filtering

1. Introduction

In this paper we are designing an interface between computer and human with the help of a camera which is used for taking the input gestures from the user. To make it more efficient and user friendly, the gestures of each user can be customized, *i.e.* each user can add his or her own gestures and assign its meaning to the gesture which makes it a lot easier for the user to interact.

In this paper, accept various hand gestures performed by user via webcam, to process its meaning, match it with the defined values in database and to display the corresponding output on the screen.

This is a very straightforward process of code in which the webcam is use to take input from the user. The input is then filtered to get only the hand gesture which is used for further processing, the remaining part is discarded as there is no use of it for further processing. After extracting the gesture from the input it is processed to calculate some basic values which are then used for comparison with those in database. If after comparison positive match is found then the corresponding meaning of that gesture is

displayed to the user. This process continues to provide results till the user exits the program.

Besides the main purpose, this technique can also be used in following fields by slightly changing the implementation:

- 3D animation: Hand movements can be easily mapped to computer space for the purpose of 3D animation.
- Communication at a distance: With the help of hand gesture we don't have to touch anything, we can communicate at a distance.
- Computer Games: Browser based games which may use hand gestures as input.
- To control mouse and keyboard using hand Gestures
- Conversion of hand movements to 3D computer space
- Using hand Gestures to interact with games
- To control Robots *etc.*

2. Previous Works

The clue is to sort computers recognize humanoid language and progress a user welcoming human computer interfaces (HCI). Creating a computer understands human gestures, speech and facial terminologies are certain steps towards it. Gestures are the non-vocally swapped evidence. An individual can execute immeasurable gestures at a period. Consequently humanoid gesture is apparent over visualization. It is a theme of pronounced attention for computer vision investigators. The venture targets to conclude humanoid gesticulations by generating an HCI. The coding of these gesticulations hooked on machine language anxieties a multifaceted programming procedure [10].

Utmost of the current hand recognition tools [2] do not assemble all the evidence or they are stationary, they produce a good view, but do not produce correct output at all situations. Therefore they work on some platforms very well but not on all the platforms. The diverse tools available for recognition of gesture are founded on the tactics vacillating from pattern recognition, image processing, statistical modeling and computer vision *etc.* In this paper our aim is to progress a scheme which can categorize sign language precisely.

2.1. Tools and Libraries

An OpenCV library to make system calls, it does not need to be re-written to implement those system calls over and over again. In addition, the behavior is provided for reuse by multiple independent programs or by tools. Here the comparison of existing tool is given in Table no.1.

2.1.1. Gesture Recognition Toolkit

GRT is a cross-platform, open-source, C++ machine knowledge public library that has been exactly intended [3] for real-time gesture gratitude.

The Gesture Recognition Toolkit has been intended to:

- be tranquil to practice and assimilate into your prevailing C++ schemes
- be companionable with any type of feeler or data input
- be tranquil to hastily train with your own gesture

The GRT topographies a huge amount of procedures that can be castoff to identify static poses, identify dynamic temporal gestures (such as a swipe or tap gesture).

2.1.2. Open Source Kinect Gesture Recognition Project, Kinect

It supports many languages such as C, C++, Python, Java *etc.* It can run on Windows, Linux platforms. It's advantages are that it can record and recognize gestures, including gestures packing, tweak able limits for well regulator of in what way the recognizer works inside and out of the box - just track the taster application, source code accessible too if you want to mark your private gestures. [4] But the main disadvantage is it requires additional hardware which increases cost is so high.

2.1.3. Js-Object Detect

It is a JavaScript library for real time object detection. It supports JavaScript, jQuery. It is browser based and multi-platform. It is compatible to stump based classifiers used by OpenCV [5]. There are some classifiers available in this library are for face, hand and eye detection. But there is no documentation available.

2.1.4. OpenCV

OpenCV is a popular open-source implementation of various Image Processing and Computer Vision functions. It contains implementation of various filters and data structures required for image processing. It has Convex Hull function built in. OpenCV has its implementation done natively; [6] therefore it cannot be used directly for web pages. It doesn't have any built-in modules for hand gesture recognition.

Table 1. Comparison of Existing Tools

Name	Language/ Platform	Advantage	Disadvantage
1. Gesture Recognition Toolkit ^[3]	Open-source, C++ Cross platform	Machine learning library	1. Only C++ language 2. It does not work well when the data you want to classify is not linearly separable
2. Kinect DTW, Open source Kinect gesture recognition project ^[4]	Many languages supported such as C, C++, python, java, <i>etc.</i> Windows, Linux	1. Recording gesture. 2. Saving gestures to database file. 3. Pinch able parameters for well regulator of how the recognizers work. 4. All of it out of the box - objective outing the sample app.	Costly requires proprietary hardware.
3. js-object detect is a JavaScript library for real-time object detection ^[5]	JavaScript, jQuery, Multiplatform, browser based	Compatible to stump based classifiers used by OpenCV. Classifiers for face, hand and eye detection are already included	1. Only two gestures available 2. No Documentation available

4. OpenCV ^[6]	C, C++, python extensions, Multiplatform	1. Various Filters and structure available 2. Covex Hull function built in data	1. No built in hand gesture recognition 2. Covex Hull requires still background
--------------------------	--	--	--

2.2. Depth Sensors

It enables users to control onscreen interaction with simple hand motions instead of a remote control, keyboard or touch screen [7]. Various depth sensors are available. Here comparisons between various depth sensors are given in Table no. 2.

2.2.1. Microsoft Kinect

It uses a VGA (640x480) video camera and QVGA (320x240) depth camera. It requires Microsoft Kinect SDK and openNI SDK library. Its current range is roughly 1.2 to 3.5 meters. It processes 30 frames per second. But it has a disadvantage that it does not work well in bright sunlight and is very high. In India its price is approximately INR 10,500/-.

2.2.2. Time of Flight

Camera works on relatively low resolution *i.e.* on 144x176. It processes 50 frames per second which is quite high.

2.2.3. Stereoscopic Cameras

It uses pair calibrated video cameras. But it has disadvantages that it has lower fidelity depth images than ToF cameras. Its advantage is that works well in bright light.

Table 2. Comparison of Depth Sensors

Method	Technology	Efficiency	Library required	Disadvantages	Advantages
1. Microsoft Kinect ^[7]	VGA (640x480) video camera and a QVGA (320x240) depth camera	Real Range is almost 1.2 to 3.5 meters.	Microsoft Kinect SDK, openNI SDK	1. Does not work well in bright sunlight 2. Price: Rs 10,500/-	30 frames per second (fps).
2. Time of Flight (ToF) cameras ^[7]	RF-modulated light sources with phase detectors or shutters which open and close at the same rate of light pulses	Dependent on the light source	Custom	Relatively low Resolution (144x176).	50 frames per second (fps).

3.Stereoscopic cameras ^[7]	Pair of calibrated video cameras	Implementation dependent	Custom	Lower fidelity depth images than ToF cameras	Work well in bright light
---------------------------------------	----------------------------------	--------------------------	--------	--	---------------------------

2.3. Feature Extraction

To find the undertaking evidence, the input gesture is expected to be non-static or stirring. Once objects passage in spatial-time space, an image structure is spawned, gesture sensor is capable to trail the stirring objects by investigative the native gray-level variations.

2.3.1. Skin Color Detection

Skin can be simply distinguished by consuming the color evidence. We use the restraint, *i.e.* $R < G < B$, to find the skin color sections which might contain a varied variety of colors, such as red, pink, brown colors. So, we will discover many provinces advance than the skin provinces. Conversely, those non-skin regions which fulfill our constriction will be barred due to there is no gesture evidence, *e.g.* orange color area will not be misidentified as hand area. Amalgamation of skin color, motion and edge. The hand gestures evidence involves of edge feature, skin color and movement

2.3.2. Edge Detection

Edge detection is functional to distinct the arm province from the hand province. There are scarcer edges on the arm province than on the palm province. A modest edge detection practice to obtain altered technique edges is used, and then the utter determined value of each pixel is select to from the edge image of *ith* frame.

2.3.3. Thresholding

Having mined the stirring object province, the thresholding on the frame variance can be functional for the mining of the likely moving province in multifaceted background. Predictable thresholding approaches, for example Ostu thresholding are not appropriate for the case of sensing wave variance. Instead, a simple thresholding method is used to mine stirring areas.

3. Design and Algorithm

Design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Algorithm design is a specific method to create a mathematical process in solving problems. Algorithm is a self-contained step-by-step set of operations to be performed.

3.1. Design [8]

The proposed solution consists of following components:

3.1.1. Hand Detection

Hand detection module primarily identifies the probable area in which the user's hand exists in the image in addition the specific rectangular area is then passed to the gesture detection system. The user enables hand detection by placing hand in forward-facing of camera at assured distance. Once the hand is detected, the user is notified with the same. The webcam access is required before we proceed towards hand detection.

3.1.2. Hand Tracking

Once a hand is perceived by the system, the hand tracking component is used to track the hand in the view of the webcam. This is required as it is necessary that the system should follow [9] hand movement continuously and also accurately to recognize various gestures.

3.1.3. Stored Text File (Database)

A text file is used to store various classifiers needed to recognize various gestures. Text file consist of many strong classifiers which in turn consist of weak classifier. Each weak classifier is set of multiple classifiers and each classifier is one of the features. These classifiers are used to recognize various gestures by the system.

3.1.4. Hand Gesture Recognition

Hand gesture gratitude component performs the actual work of recognizing the gestures achieved by the user. This software shall make skin color discovery and filter obtainable objects that do not contain the color of skin. By filtering object of non-skin colored, the system can then routine its residual assets and emphasis on hand discovery and gesture acknowledgements. This also allows the system to pinpoint likely sites of user's hands.

It draws a convex hull and convexity defects around the detected object. The recognition system then matches the movement of hand with stored classifiers and tries to identify the gesture. If complete set of simple classifiers is successfully detected, then the system considers the respective gesture is performed by user.

3.1.5. Event Handling

Once the system recognizes the gestures, the system notifies the user system with the action to be performed with respect to the gesture. The actions on gestures are predefined by the system and stored in text file. So, particular event takes place when user performs the particular gesture. The event handling is demonstrated with the help of displaying the meaning associated with particular gesture.

3.2. Algorithm

The proposed solution provides a novel approach for the problem of gesture recognition.

- Step 1. Take input from webcam
- Step 2. Capture image from the webcam
- Step 3. Apply filters on image such as gray scale conversion
- Step 4. Calculate values from grayscale image after applying threshold
- Step 5. Repeat step 4 until all gesture are taken from the user
- Step 6. Take text meaning of each gesture from the user
- Step 7. Execute the real time comparison take the gestures from the user
- Step 8. Draw the convex hull and contours on the detected object
- Step 9. Compare the values obtained from the hull and contours and match it with database values
- Step 10. Is match is found then display its corresponding result and store its meaning in list, If match not found goto step 7
- Step 11. If Exit=True stop execution of program, execute the statement in list, else goto step 7

3.3. System Architecture

System architecture is basically divided into five major subsystems as shown in Figure 1 which are as stated below:

3.3.1. Image Capture

The function of image capture is for hand detection through webcam. After allowing webcam access, user positions his/her hand at some distance from screen for detection.

3.3.2. Object Detection

Feature uncovering stage is worried with the discovery of structures which are used for the approximation of restrictions of the preferred gestural model. In the detection procedure it is first essential to restrict the gesture. Once the gesture is restricted, the desired set of structures can be performed.

3.3.3. Gesture Recognition

Once the hand is detected, the convex hull is applied on the detected object and it is matched with the predefined shapes. If a match is found, respective meaning of the gesture from the database is selected and result is displayed

3.3.4. Hand Tracking

Tracker is applied to increase the accuracy and to reduce the false positives in the system. The tracker is applied only on the area returned by previous object detection. It reduces the region of interest (ROI) and increases recognition rate. Also the false positives can be reduced and accuracy is increased.

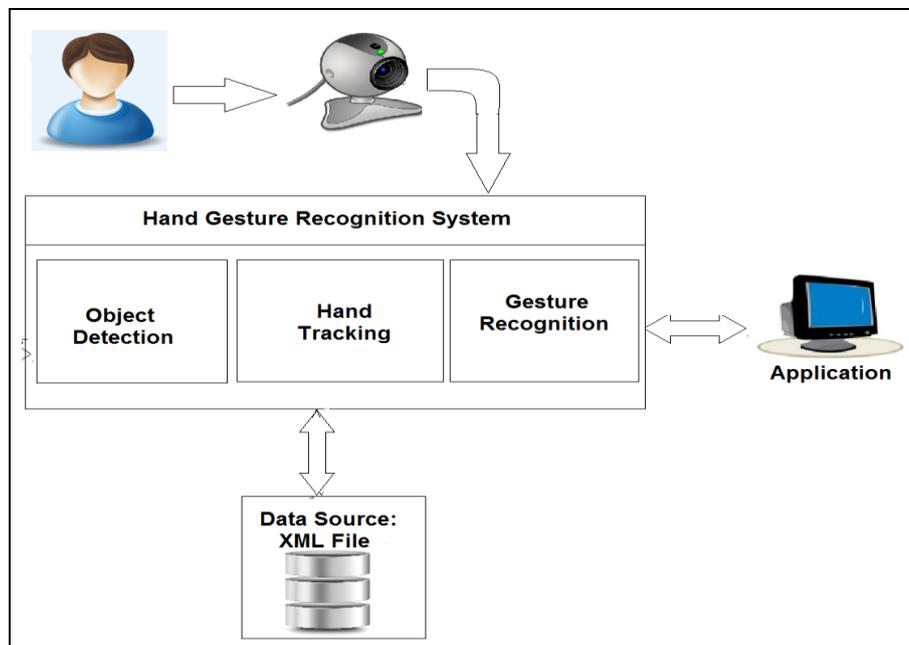


Figure 1. System Architecture

3.4. Set Representation of the System

$S = \{I, O, P, Sc, F\}$

Where,

S -> System

I -> Input,

O -> Output,

P -> Processes,

Sc -> Success case,

F -> Failure case

$I = \{\text{frames captured by camera}\}$

$O = \{\text{actions performed according to hand gesture}\}$

$P = \{\text{elimination of noise, edge detection, segmentation, feature measurement, scene analysis}\}$

$Sc = \{\text{detection of the hand gesture, according to hand gesture performing the appropriate action}\}$

$F = \{\text{hand detection is not done correctly, detection of wrong hand gesture, doing the wrong action for particular hand gesture}\}$

3.5. Software Interfaces

Software interface are the languages and codes that the applications use to communicate with each other and with the hardware. Different software interfaces provides different way of communication.

3.5.1. OpenCV

OpenCV is a popular open-source implementation of various Image Processing and Computer Vision functions. It contains implementation of various filters and data structures required for image processing. It has Convex Hull function built in. OpenCV has its implementation done natively; therefore it cannot be used directly for web pages. It doesn't have any built-in modules for hand gesture recognition.

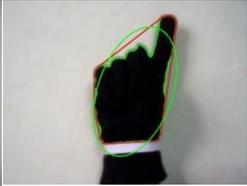
3.5.2. Python

Python is a high-level, inferred, collaborative and object-oriented scripting language. Python was premeditated to be extremely legible which practices English keywords recurrently whereas further languages habit punctuation and it has fewer syntactical creations than additional languages.

4. Result and Analysis

Few gestures were stored and compared for testing purpose. Table no 3 is the table which shows the results of the test.

Table 3. Result and Efficiency

G E S T U R E N O	GESTURE	CONVEX HULL AND CONTOUR	EXPEC TED RESUL T	ACTUAL RESULT	ACCU RACY	TOTA L NO OF ATTE MPTS
1			Zero	Zero	90%	100
2			One	One	90%	100
3			Two	Two	80%	100
4			Three	Three	90%	100
5			Four	Four	80%	100
6			Five	Five	90%	100
7			Hey	Hey	60%	100

8			See you	See you	60%	100
9			Over There	Over There	70%	100

4.1. Analysis Parameters

In order to find out the recital and feasibility of the planned gesture gratitude system subsequent testing and analysis parameters could be considered.

4.1.1. Stoutness

In the real-world, visual evidence could be very ironic, loud, and inadequate, due to varying brilliance, clutter and dynamic backgrounds, sealing, *etc.* Vision-based systems should be user liberated and robust in contradiction of all these influences.

4.1.2. Scalability

The Vision-based collaboration system should be simply altered to dissimilar scales of applications. For *e.g.* the core of Vision-based interface should be the matching for desktop environments, Sign Language Recognition, robot navigation and also for VE.

4.1.3. Computational Efficiency

Generally, Vision based interaction often needs real-time systems. The vision and learning methods/procedures used in Vision-based interaction should be actual as well as cost effectual.

4.1.4. User's Tolerance

The faults or mistakes of Vision-based interface should be endured. When a error is made, it should not suffer much loss. Users can be asked to recurrence some movements, instead of letting the computer make more incorrect results.

The computer vision techniques used in the application for manipulation of objects in virtual environment have been implemented in C++ with the use of Open CV Library. The virtual objects (front end) have been intended by OpenGL library. The hardware necessities of the request to be applied include computer with 1.99 GHz processor. The web cam used in the trial setup captures image arrangements at the resolution of 320x240. Practical trials show that our application is implemented well in surroundings with little noises (*i.e.*, presence of objects whose color is comparable to human skin) and with the composed lightning disorder. Here are some advantages and disadvantages mentioned in Table no. 4:

Table 4. Advantages and Disadvantages of Proposed System

Advantages	Disadvantages
Hardware and software requirements are low.	Light background needed for better output.
Easy to implement.	Processing speed depend upon quality of processor.
Product cost is low.	Gesture must be stationary.

5. Conclusion

An effective sign language recognition system could provide an chance for the deaf to interconnect with non-validation publics without the want for an interpreter. It could be used to engender dialogue or text making the deaf more sovereign. Regrettably there has not been any system with these competences so far [10]. From the proposed system we can recognize hand gestures and process them to display the corresponding output and this can be done with a mid-end computer specification.

This system provides the user to add his custom hand gestures and their meaning to the database which makes the system more user friendly. The system provides medium accuracy and is able to provide high accuracy if the user is able to provide the most similar hand gestures as stored in the database.

6. Future Work

The proposed system does not use any depth detection measures. It only works in a 2D plane. Introduction of a third dimension will add more intuitiveness to the system. The system cannot be used for non-browser based applications. Implementing the code natively will bring the gesture recognition functionality to the operating system level.

References

- [1] R. Shrivastava, "A Hidden Markov Model based Dynamic Hand Gesture Recognition System using OpenCV", IEEE International Conference on Advance Computing (IACC), (2013), pp. 947-950.
- [2] J. Suarez and R. R. Murphy, "Hand Gesture Recognition with Depth Images: A Review", IEEE International Symposium on Robot and Human Interactive Communication, (2012), pp. 37-39.
- [3] V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, (1997), pp. 677-695.
- [4] D. Shah, "Visual gesture recognition", IEEE -Vision, Image and Signal, vol. 141, no. 2, (1994), pp. 101-106.
- [5] R. Cipolla and A. Pentland, "Computer Vision for Human Machine Interaction", Cambridge University Press, (1996), pp. 03-07.
- [6] C. Vogler and D. N. Metaxas, "Parallel hidden markov models for American sign language recognition", Computer Vision, 1999. IEEE International Conference on Computer Vision, vol. 1, (1999), pp. 116-122.
- [7] N. Gupta, A. Jain, T. B. Patil and S. C. Sawant, "Comparative Survey and Analysis Of Hand Gesture Recognition On Various Gesture Recognition Technologies And Techniques", International Journal of Computer Science And Technology, vol. 7, no. 2, (2016), pp. 107-109.

- [8] M. Allister, G. McKenna and S. J. Ricketts, "Hand tracking for behavior understanding", *Image and Vision Computing*, vol. 20, no. 12, (2002), pp. 827-840.
- [9] W. T. Freeman and M. Roth, "Orientation histograms of hand gesture Recognition", *IEEE International Workshop on Automatic Face and Gesture Recognition*, Zurich, (1995), pp. 14-19.
- [10] S. Meena and S. Ari, "A Study on Hand Gesture Recognition Technique", *Department of Electronics and Communication Engineering National Institute of Technology, Rourkela*, pp. 6-13.
- [11] W. Jiangqin, G. Wen, S. Yibo, L. Wei and P. Bo, "A simple sign language recognition system based on data glove", *International Conference on Signal Processing*, Beijing, China, vol. 2, (1998), pp. 1257-1260.