

Hand Gesture Recognition for Mute People

Anupama H S¹, Dr. Cauvery N K², Dr. Lingaraju G M³

Asst Professor, CSE, RV College of Engineering, Bangalore, India¹

Professor, ISE, RV College of Engineering, Bangalore, India²

Professor, ISE, MSRIT, Bangalore, India³

Abstract: More than 500 million people of the world suffer from some physical, sensory or mental disability. Often their lives are handicapped by physical and social barriers which hamper their full participation from society and the enjoyment of equal rights and opportunities. Many mute people use sign language. Sign language uses gestures instead of sound to convey meaning, simultaneously combining hand shapes, orientations and movement of the hands, arms or body and facial expressions to express fluidly a speaker's thoughts. Signs are used to communicate words and sentences to audience. This paper focuses on developing an algorithm for recognition of hand gestures with reasonable accuracy, where the input to the pattern recognition system will be given from the hand. It recognizes the pattern and the display the pattern in the form of the text. The info giving by camera are collected and stored in the database. The exact meaning of the information is matched with the samples stored previously in the database and is printed. The image is processed considering the parameters like the number of fingers used, the angles between them and hsv value of that which is matched with the saturation image and then the information is displayed in the form of text.

Keywords: Hand Gestures, Computer Vision, Sign Languages, Open CV, and Mute people.

1. INTRODUCTION

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. A theme in the development of this field has been to duplicate the abilities of human vision by electronically perceiving and understanding an image. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory. Computer vision has also been described as the enterprise of automating and integrating a wide range of processes and representations for vision perception. [1,2,4]

Applications range from tasks such as industrial machine vision systems which, say, inspect bottles speeding by on a production line, to research into artificial intelligence and computers or robots that can comprehend the world around them. The computer vision and machine vision fields have significant overlap. Computer vision covers the core technology of automated image analysis which is used in many fields. Machine vision usually refers to a process of combining automated image analysis with other methods and technologies to provide automated inspection and robot guidance in industrial applications. As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models to the construction of computer vision systems.

2. AMERICAN SIGN LANGUAGE (ASL)

American Sign Language (ASL, also Ameslan) is the dominant sign language of the Mute community in the United States, in the English-speaking parts of Canada, and in parts of Mexico. "It is a manual language or visual language, meaning that the information is expressed not with combinations of sounds but with combinations of handshakes, palm orientations, movements of the hands, arms and body, and facial expressions." [3,5]

The main project task is to produce a gesture recognition system, the system will be able to recognize the ASL gesture using web cameras. In the real world, it has practical usage, if the mute demonstrates the hand posture in front of the web-camera, the system will detect which letter it is in the ASL; If we know which letter it is, we will understand which word several gesture represents, it will help mute people to communicate with normal people more easily

3. SOFTWARE DESIGN

The software design for implementing the hand gesture intelligent is shown in Fig 1. In this project a real time image is captured from the USB camera and a detected frame is saved. All the strong Haar classifiers are loaded into the system. Then the captured frame is scanned from left to right and from top to bottom using a sub-window which consists of the Haar classifier. If the hand gesture is detected, bounding box is drawn around the gesture. Each classifier that is loaded into the system is assigned an integer to classify different gestures. Depending on the gesture detected appropriate integer is displayed in the result window. [1,6,7,2]

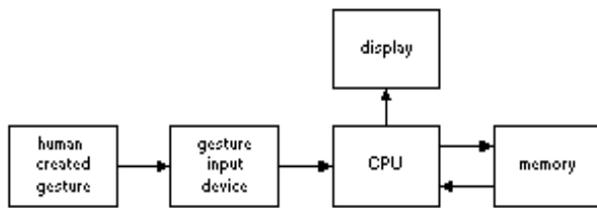


Figure 1: Architecture of Hand Gesture

4. HAND GESTURE RECOGNITION

A. Introduction

After conducting an extensive literature survey, conclusions were drawn as to what would be the right path to make this attempt of implementing a real-time hand gesture recognition algorithm. Apart from Hidden Markov Models, using multiple cameras, using techniques of segmentation, filtering, representation, and classification, comparing shape descriptors, and using Quadruple Visual interest Point strategy, it was decided to implement an algorithm that would use Haar-like features and Adaptive Boosting machine learning algorithm for detection and recognition of hand gestures. These two methodologies, when coupled together, form a very efficient real-time system.[1,2,3]

B. Tools and Preparations

For our real-time implementation, we used the open source library, OpenCV with GCC compiler. The library is mainly written in C, which makes it portable to some specific platforms such as digital signal processors. So, in the future implementing the algorithm on hardware will not be tough. OpenCV does have pre-defined C and C++ functions which can be made use of extensively in the source code, and some of those functions in the library even optimize the operations depending on the situation. These features made OpenCV an obvious chance of our implementation tool.

C. OpenCV

OpenCV is an open source computer vision library available from the library is written in C and C++ and runs under Linux, Windows and Mac OS X. There is active development on interfaces for Python, Ruby, Matlab, and other languages. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. OpenCV is written in optimized C and can take advantage of multicore processors. For further automatic optimization on Intel architectures, Intel's Integrated Performance Primitives (IPP) libraries [IPP] can be used, which consist of low-level optimized routines in many different algorithmic areas. OpenCV automatically uses the appropriate IPP library at runtime if that library is installed. One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. The OpenCV library contains over 500 functions that span many areas in vision, including factory product inspection, medical imaging, security, user interface, camera calibration, stereo

vision, and robotics. Because computer vision and machine learning often go hand-in-hand, OpenCV also contains a full, general-purpose Machine Learning Library (MLL). This sub-library is focused on statistical pattern recognition and clustering. The MLL is highly useful for the vision tasks that are at the core of OpenCV's mission, but it is general enough to be used for any machine learning problem.

D. OpenCV Structure and Content

OpenCV is broadly structured into five main components, four of which are shown in Fig 2. The CV component contains the basic image processing and higher-level computer vision algorithms; ML is the machine learning library, which includes many statistical classifiers and clustering tools. HighGUI contains I/O routines and functions for storing and loading video and images, and CXCore contains the basic data structures and content.

Figure 2: Architecture Open CV

E. Image capture and extraction

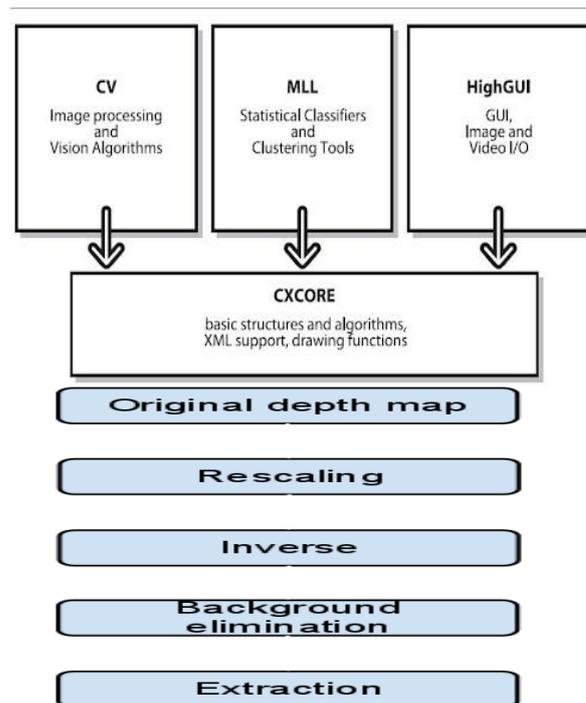


Figure 3: Steps involved in extraction

The image is captured and extracted as shown in Figure 3.

F. Original depth map

The first step is to obtain the image from the depth camera. With the OpenCV, Each frame is converted into a matrix in uint8 gray scale. However, the raw image is barely recognizable due to low color contrast, and thus the whole image is dark.

G. Rescaling

To improve the raw image, we rescaled it. By adjusting the scale factor, we make the sensitive range to be from the

minimum depth. This maximum depth should reach some point at the user's arm. The hand is in the sensitive range, so it's in gray scale, but the body and the background will be considered as the same, which is the maximum of depth, and thus not visible.

H. Background elimination

The white background and the body are not necessary for extraction, so we performed a digital-negative operation on the image to make the image more optimal for subsequent steps. The shadows around the hand and the body, due to the positional disparity between the HSV camera and the depth sensor, can severely affect the performance of the recognition, so we have to eliminate them. Based on the previous step, the shadow at this point is white, while the background is black. The gray part is the hand, which is in the sensitive range. We can eliminate the shadows by thresholding the gray level.

I. Extraction

To perform extraction, we have to improve our image further. We captured the top point of the palm of the hand, and kept the pixels which represent the depth up to 24 units from that point. Otherwise, they will be eliminated. All pixels in this range, will be converted into white (intensity thresholding). Finally we extracted a hand shape image for recognition.

J. Object Detection & Tracking using Color

Object detection and segmentation is the most important and challenging fundamental task of computer vision. It is a critical part in many applications such as image search, image auto-annotation and scene understanding. However it is still an open problem due to the complexity of object classes and images.

The easiest way to detect and segment an object from an image is the color based methods. The colors in the object and the background should have a significant color difference in order to segment objects successfully using color based methods.

HSV color space consists of 3 matrices, 'hue', 'saturation' and 'value'. In OpenCV, value range for 'hue', 'saturation' and 'value' are respectively 0-179, 0-255 and 0-255. 'Hue' represents the color, 'saturation' represents the amount to which that respective color is mixed with white and 'value' represents the amount to which that respective color is mixed with black.

to rotation and lighting variations. Some of the key point has been experienced during the project development are:

- If the weak classifier is big enough, the strong classifier produced will have extreme low error rate.
- Increase the amount of positive sample size will improve the detection accuracy.
- The sample size for create sample is same as training sample.
- The Haar-training takes weeks, if it is interrupt, it is able to restart and carry forward the training.

Here we used the HSV color of yellow. The hsv color of yellow is used to detect the hand. When we run the code two windows are displayed. In first window yellow color is turned into white and background is turned into black color.. In second window hand is moved according to that the color is scribbled yellow color glove is detected and tracked.. Over all description of the image capture and displayed is shown with a flow chart as shown in Figure 4.

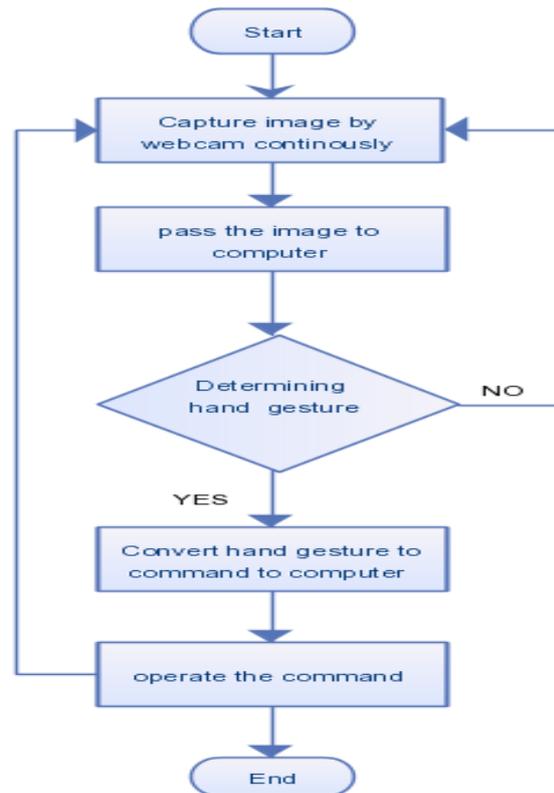


Figure 4: Flow Chart of the hand gesture

5. CONCLUSIONS

Hand gesture recognition is a very challenging and interesting task in computer vision, and the technology about gesture recognition is getting mature in the real world application. For the hand detection and gesture recognition, the famous AdaBoost algorithm which is proposed by Viola-Jones is used to create a Haar-like feature classifier. The final cascade of classifiers is robust

REFERENCES

1. http://www.massey.ac.nz/~albarcza/ResearchFiles/DaKuanCui_PGDip_2009.pdf
2. <http://research.microsoft.com/en-us/um/people/awf/bmvc02/project.pdf>
3. <http://se.cs.ait.ac.th/cvwiki/opencv:tutorial:haartraining>
4. <http://note.sonots.com/SciSoftware/haartraining.html>
5. <http://www.andol.info/hci/1661.htm#comment-7271>
6. <http://www.technolabsz.com/2011/08/how-to-do-opencv-haar-training.html>
7. <http://areshopencv.blogspot.in/2011/07/build-cascade-of-boosted-classifiers.html>