

Hand Gesture Controls for Digital TV using Mobile ARM Platform

Ajit S. Gundale¹ Pallavi G. Potadar²

^{1,2} Electronics Department

Walchand Institute of Technology Solapur, Maharashtra (INDIA)

Abstract-- This paper presents a new approach for controlling digital television using a real-time camera. Proposed method uses a camera, a mobile ARM platform and computer vision technology, such as image segmentation and gesture recognition, to control TV operations such as changing channels, increasing or decreasing volume etc. For this we have used an ARM based mobile platform with OMAP processor. For processing the images we implemented the code using OpenCV library. Hand detection is one of the important stages for applications such as gesture recognition and hand tracking. In this paper, it proposes a new method to extract hand region and consequently the fingertips from color images.

Keywords: Codeblocks, Fingertips detection, OMAP4430, OpenCV, Panda board.

I. INTRODUCTION

Gesture recognition is an art of computer science and language technology human gestures interpreted using various mathematical algorithms. Current focuses in the field include emotion recognition from the face and hand gesture recognition. There exist many techniques that use cameras and computer vision algorithms to interpret sign language.

In this work, we are using hand gestures to control the Television operations such as increasing or decreasing the volume, changing the channels. In this project camera is connected to the ARM board through camera expansion connector. The image captured by camera will be processed to recognize the gestures.

The processing of gestures will be done using mobile ARM platform. After processing the gestures the manipulated signals will be sent to the Television set controller to operate according to gestures.

Following flowchart shows the sequence of processing.

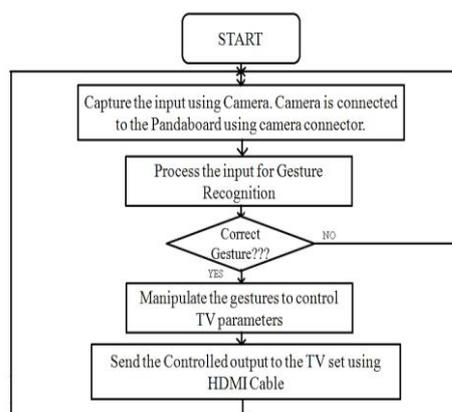


Figure 1: Overall Flowchart for the project

II. ARM BASED PLATFORM

The images captured by the camera should be processed very fast to provide real time interaction between user and television. For this purpose along with low cost we think to use ARM based Processors. Some of the reasons for the proliferation of ARM-based processors include: low cost, low-to-very-low power consumption, decent processing power, and open development environment.

III. OPENCV

OpenCV library [1] [2] is now playing a vital role in the Computer Vision field as it provides reduced cost and preparation time of computer vision research environment... One of the 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 image and video processing.

IV. METHODOLOGY

Gesture and voice-activated products will eventually migrate into all of our appliances and electronic devices making mice and remotes obsolete. We'll be able to walk into a room and adjust the lights, turn on the stereo and browse the Internet simply by speaking commands or waving our hand.

Vision-based user interfaces enable natural interaction modalities such as gestures. Such interfaces require computationally intensive video processing at low latency.

We are trying to demonstrate an application that recognizes gestures to control TV operations. In vision based hand gesture recognition system, the movement of the hand is recorded by camera(s).

In this work we implement the following,

- 1) Interfacing of camera or sensor to the USB port of the ARM mobile platform through camera connector.
- 2) Analysis the images those are captured by camera to find the specific gesture. This can be done by the following the steps as below

Step: 1 Segmenting image into hand area and background based on the HSV values of the skin

Step: 2 Removing the noise from image. The process is called De-noising.

Step: 3 Computing the center and size of the hand

Step: 4 Finding and counting the tips of the hand region

Step: 5 Finding the specific operation on the basis of figure count

3) Interfacing of ARM mobile platform to the modern digital Television set to control the Television set. The detailed block diagram of the prototype is shown below In figure 2.

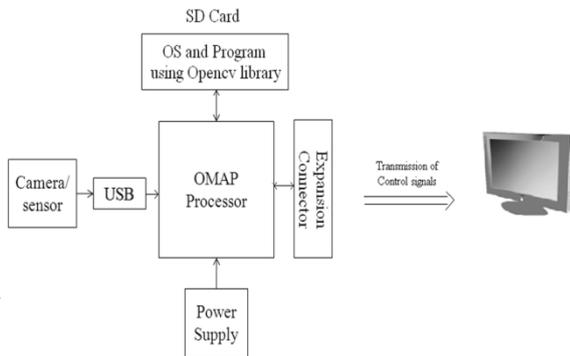


Fig. 2: Block Diagram

The input images captured by the camera are processed to detect the gestures is an ARM mobile platform. This input is processed using functions in OpenCV library. Block diagram shows camera sensor, power supply, and display and SD card. The operating system and application program reside in SD card.

V. HAND GESTURE RECOGNITION

The system order is shown in Figure 5 [3]. Initially, when we will get an image from the camera, we convert the color space RGB to YCbCr.

Then, we will define a range of colors as 'skin color' and will convert these pixels to white; all other pixels will be converted to black. Then, we will compute the centroid of the dorsal region of the hand. After we identify the hand, we will find the circle that best fits this region and multiply the radius of this circle by some value to obtain the maximum extent of a 'non-finger region'. From the vertex and center distance, we will obtain the positions of the active fingers

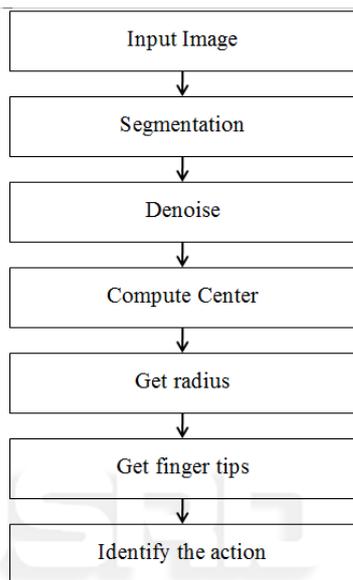


Fig. 3: An overview of our hand gesture recognition. Input image converted to binary image to separate hand from background. Center of hand and computed calculated radius of hand found.

A. Input Image

Camera or sensor will be used to capture the images when the user wants to control the TV. The limiting distance may depend on FOV (Field Of View) of camera.

B. Segmentation

Next, we need to separate the hand area from a complex background. It is difficult to detect skin color in natural environments because of the variety of illuminations and skin colors. To get better results, we converted from RGB color space to YCbCr color space, since YCbCr is insensitive to color variation. Generally, for skin Y, Cr and Cb values are in the range of 0 to 255, 77 to 127, and 133 to 173 respectively.

C. Deleting noise

Using this approach, we cannot get a good estimate of the hand image because of background noise. To get a better estimate of the hand, we will need to delete noisy pixels from the image. We can use an image morphology algorithm that performs image erosion and image dilation to eliminate noise [4].

Erosion trims down the image area where the hand is not present and Dilation expands the area of the Image pixels which are not eroded.

D. Finding center and the size of the hand

After segmenting hand region and background, we can calculate the center of the hand with the following equation:

$$\bar{x} = \frac{\sum_{i=0}^k x_i}{k}, \quad \bar{y} = \frac{\sum_{i=0}^k y_i}{k}$$

Where x_i and y_i are x and y coordinates of the i pixel in the hand region, and k denotes the number of pixels in the region. After we will locate the center of the hand, we can compute the radius of the palm region to get hand size.

E. Finding finger tip

The finger count is determined as in [5]. The centroid of the segmented binary image of the hand is calculated. Then the length of the largest active finger is found by drawing the bounding box for the hand. The centroid calculated is made as the Centre and the value of radius is the length of the largest finger multiplied by 0.7 [4]. With the centroid as Centre and length of the largest finger multiplied with 0.7 as radius a circle is drawn to intersect with the active fingers of the hand. If a finger is active then it intersects with the circle.

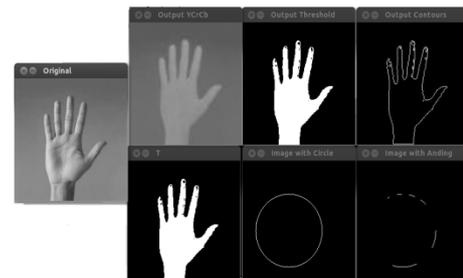


Fig. 4: Example showing the result of finger count algorithm.

We assigned control operation to the number of fingers as shown in the table below. Those control signals are transmitted using IR LED connected to ARM platform.

Finger count	Functions
0	Turn OFF
1	Turn ON
2	Previous Channel
3	Next Channel
4	Decrease Volume
5	Increase Volume

Table 1: Gestures and Their Functions

VI. PERFORMANCE EVALUATION

In this method we used Panda board [6] as a mobile ARM platform. The heart of Panda Board is the OMAP4430 processor. The OMAP4430 high-performance multimedia application device is based on enhanced OMAP™ architecture. The architecture is designed to provide best-in-class video, image, and graphics processing sufficient to various applications.

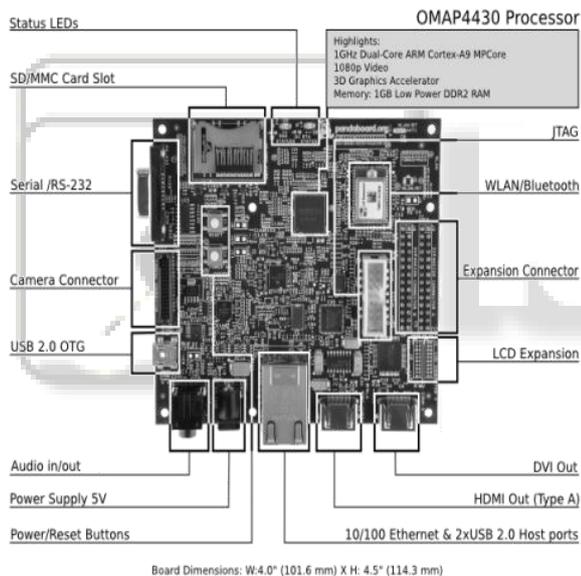


Fig. 5: OMAP4430 Panda Board

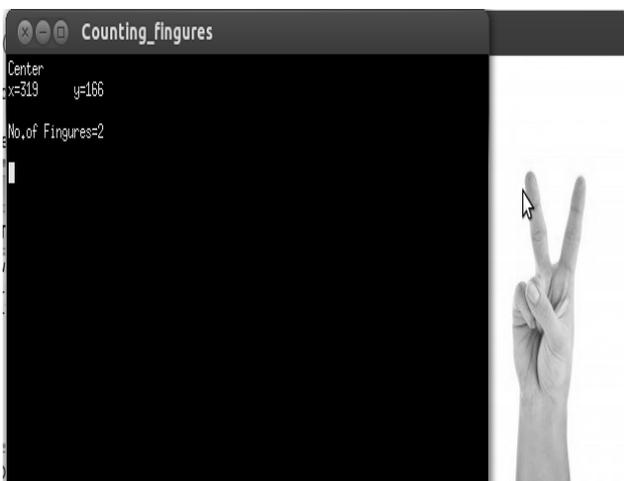


Fig. 6: Counting Fingers



Fig. 7 : Counting Fingers

The device supports the following functions:

- 1) Streaming video up to full high definition (HD) (1920 × 1080 p, 30 fps)
- 2) 2-dimensional (2D)/3-dimensional (3D) mobile gaming
- 3) Video conferencing
- 4) High-resolution still image (up to 16 Mp)

The Platform also includes connectors that can be used for additional functionality and/or expansion purposes. Those are,

- 1) Camera Connector
- 2) LCD Expansion Connectors
- 3) Generic Expansion Connectors
- 4) Composite Video Header

Camera is connected to Panda board through USB. The output control signals are transmitted using IR LEDs.

With this setup Fig. 6, 7, 8, 9 and 10 are results are obtained.



Fig. 8: Counting fingers



Fig. 9: Counting fingers



Fig. 10: Counting Fingers

VII. RESULTS

Table shows the performance of our approach comparatively with that of all other approaches on their test results using their own database test images in terms of recognition time, accuracy, resolution, number of tested images, background.

Reference	No. of Postures	Recognition Time	% Accuracy	No. of Test Image	Background
[Chung09]	15	0.4	94.89	30	Wall
[Fang07b]	6	0.09 – 0.11	93.8	195-221	Cluttered
[Yun09]	3	0.1333	96.2	130	Different
[Chen07]	4	0.03	90.0	100	White wall
[Ren10]	8	0.06667	96.9.	300	Not discussed
[Marcel99]	6	Not discussed	93.4	57-76	Wall
[Marcel99]	6	Not discussed	76.1	38-58	Cluttered
Proposed Method	6	0.05089	96%	50	Wall

Table. 2: Comparison between various approaches

After successful recognition of gestures, the remote control codes are transmitted using IR LED which is connected to Panda board.

VIII. CONCLUSION

In today's digitized world, processing speeds have increased dramatically, with computers being advanced to the levels where they can assist humans in complex tasks. From the performance analysis we can see that gestures are recognized in only 50 mSec with 96% accuracy. Technologies developed based on gesture are now really affordable and converged with familiar and popular technologies like TV, large screen. We will also conclude that a lot of applications can be created by using just the finger count of a hand provided we will be able to distinguish between the fingers.

The fast and efficient algorithm for gesture recognition will be cost effective if used with mobile ARM platform.

REFERENCES

[1] "Learning OpenCV" by Gary Bradski and Adrian Kaehler
 [2] Willow Garage, <http://www.willowgarage.com>.
 [3] Hojoon Park Department of Computer Science Brown University, Providence, RI, USA, hojoon@cs.brown.edu

[4] Computer vision based mouse, A. Erdem, E. Yardimci, Y. Atalay, V. Cetin, A. E. Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASS). IEEE International Conference
 [5] Ram Rajesh J, Sudharshan R, Nagarjunan D, Aarthi R, "Remotely controlled PowerPoint presentation navigation using hand gestures", and unpublished.
 [6] <http://pandaboard.org>.
 [7] [Chung09] W. Chung, X. Wu, and Y. Xu, "A real-time hand gesture recognition based on Haar wavelet representation," in Proc. IEEE Int. Conf. Robot. Biomimetics, 2009, pp. 336–341
 [8] [Fang07b] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in Proc. IEEE Int. Conf. Multimedia Expo, 2007, pp. 995–998.
 [9] [Chen07] Q. Chen, N. Georganas, E. Petriu, "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features", IEEE Instrumentation and Measurement Technology Conference Proceedings, IMTC 2007
 [10] [Ren10] Y. Ren and C. Gu, "Real-time hand gesture recognition based on vision," in Proc. Edutainment, 2010, pp. 468–475.
 [11] [Yun09] L. Yun and Z. Peng, "An automatic hand gesture recognition system based on Viola-Jones method and SVMs," in Proc. 2nd Int. Workshop Comput. Sci. Eng., 2009, pp. 72–76.
 [12] [Marcel99] S. Marcel, "Hand posture recognition in a body-face centered space," in Proc. Conf. Human Factors Comput. Syst. (CHI), 1999, pp. 302–303.