

# Gesture Control for TV Using ARM based Mobile Platform

A.S. Gundale and Pallavi G. Potadar

**Abstract---** Many TV manufacturers are trying to embed technology into their TV sets enabling people to use hand gestures and voice recognition to adjust the volume, change channels and switch devices. Obviously currently these technologies are expensive. In this work we are developing a system in which the Television is controlled using hand gestures. For gesture recognition we will be incorporating OpenCV Libraries. OpenCV is Open Source Computer Vision Library that includes image processing functions. All the processing will be done using ARM based platform like PandaBoard which is an OMAP4430 platform designed to provide access to as many of the powerful features of the OMAP4430 Multimedia Processor as possible, while maintaining a low cost.

**Keywords---** Gestures, OMAP4430, Open CV, Panda Board

## I. INTRODUCTION

**G**ESTURE recognition is an art of computer science and language technology human gestures interpreted using various mathematical algorithms. Gestures can originate from any bodily motion or state but commonly originate from the face or hand.

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.

Gesture recognition is useful for processing information from humans which is not conveyed through speech or type. As well, there are various types of gestures which can be identified by computers.

Applications of Gesture recognition:

- Sign language recognition. Just as speech recognition can transcribe speech to text, certain types of gesture recognition software can transcribe the symbols represented through sign language into text[1].
- For socially assistive robotics. By using proper sensors worn on the body of a patient and by reading the values from those sensors, robots can assist in patient rehabilitation. The best example can be stroke rehabilitation.

- Control through facial gestures. Controlling a computer through facial gestures is a useful application of gesture recognition for users who may not physically be able to use a mouse or keyboard. Eye tracking in particular may be of use for controlling cursor motion or focusing on elements of a display.
- Alternative computer interfaces. Foregoing the traditional keyboard and mouse setup to interact with a computer, strong gesture recognition could allow users to accomplish frequent or common tasks using hand or face gestures to a camera[2][3][4][5][6].
- Immersive game technology. Gestures can be used to control interactions within video games to try and make the game player's experience more interactive or immersive.
- Virtual controllers. For systems where the act of finding or acquiring a physical controller could require too much time, gestures can be used as an alternative control mechanism. Controlling secondary devices in a car, or controlling a television set are examples of such usage[7].
- Affective computing. In affective computing, gesture recognition is used in the process of identifying emotional expression through computer systems.
- Remote control. Through the use of gesture recognition, "remote control with the wave of a hand" of various devices is possible. The signal must not only indicate the desired response, but also which device to be controlled [8][9][10].

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 platform like Panda board. After processing the gestures the manipulated signals will be sent to the Television set controller to operate according to gestures

Computer Vision is a scientific field which attempts to provide a sight to the machine. This scientific field has expanded rapidly in recent years. Among researchers this growth is because of many improvements of vision algorithms and among the computer vision hobbyists this is due to the cheaper hardware components and processing power.

Open CV library plays a great role in the Computer Vision field as it helps greatly to reduce cost and preparation time of computer vision research environment needed by university

A.S. Gundale, Electronics Department, Walchand Institute of Technology Solapur, Maharashtra (India).

Pallavi G. Potadar, Electronics Department, Walchand Institute of Technology Solapur, Maharashtra (India). E-mail: pallavipotadar@gmail.com

students, hobbyists and professionals. OpenCV also provides a simple to use functions to get the work done in a simple, effective and elegant manner.

Following flowchart shows the sequence of processing.

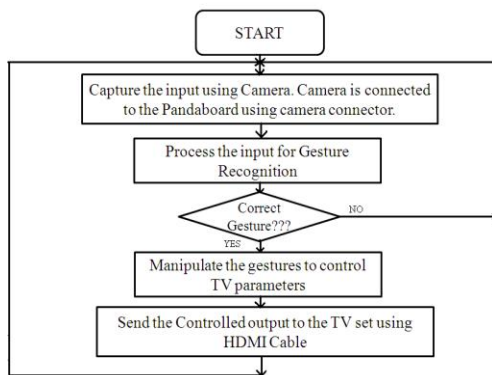


Figure 1: Overall Flow Chart for the Project

### 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.

The OMAP 4 platform includes applications processors, a comprehensive software suite, power management technology. The heart of PandaBoard 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.

The device supports the following functions:

- Streaming video up to full high definition (HD) (1920 × 1080 p, 30 fps)
- 2-dimensional (2D)/3-dimensional (3D) mobile gaming
- Video conferencing
- 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,

- Camera Connector
- LCD Expansion Connectors
- Generic Expansion Connectors
- Composite Video Header

### III. 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,

- Interfacing of camera or sensor to the USB port of the ARM mobile platform like Pandaboard through camera connector.
- Analysis the images those are captured by camera to find the specific gesture. This can be done by the following steps as below
  1. Segmenting image into hand area and background based on the HSV values of the skin
  2. Removing the noise from image. The process is called De-noising.
  3. Computing the center and size of the hand
  4. Finding and counting the tips of the hand region
  5. Finding the specific operation on the basis of figure count
- Interfacing of ARM mobile platform to the modern digital Television set through HDMI port to control the Television set.

The block diagram of proposed model is shown below.

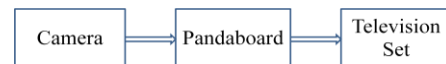


Figure 2: Block Diagram

The input images captured by the camera are processed to detect the gestures using Panda board which is an ARM mobile platform.

The detailed block diagram of the prototype is shown below.

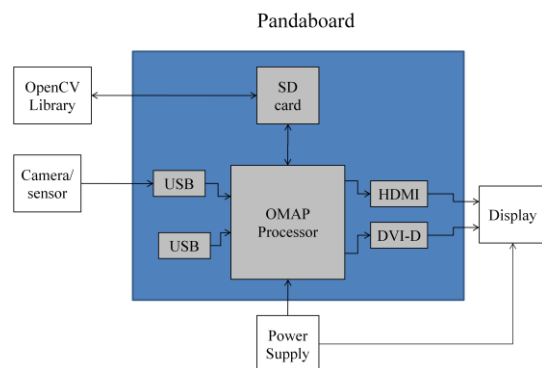


Figure 3: Detailed Block Diagram

This input is processed using functions in OpenCV library. Block diagram shows camera sensor, power supply, display and SD card. The operating system and application program reside in SD card. OMAP processor works with 5V power

supply. The digital output is provided to the TV display over HDMI cable while analog output is provided over DVI-D.

#### IV. HAND GESTURE RECOGNITION

The system order is shown in Figure 4[11]. 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 binary image of the hand, we will get vertices of the convex hull of each finger. From the vertex and center distance, we will obtain the positions of the active fingers

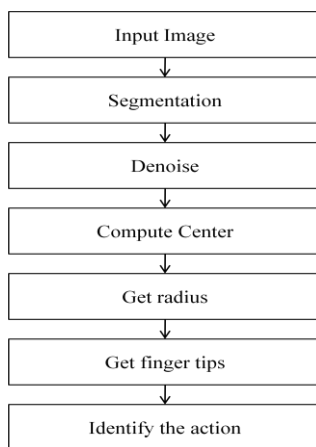


Figure 4: An Overview of our Hand Gesture Recognition

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. So, we will need to carefully pick a color range. To get better results, we converted from RGB color space to YCbCr color space, since YCbCr is insensitive to color variation. The conversion equation is as follows.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.29900 & 0.587000 & 0.114000 \\ -0.168736 & -0.331264 & 0.500000 \\ 0.500000 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where Y is luminance, Cb is the blue value and Cr is the red value. From this information, we will detect skin color by selecting a particular color range from the Cb and Cr values. Generally for skin Y, Cr and Cb values are in the range of 0 to 255, 77 to 127, and 133 to 173 respectively.

Then we loop over all the image pixels, changing pixels within the skin color range to 0, and all others to 255. Hence, we obtain a binary image of the hand. Figure 5 shows the results.



Figure 5: Example Showing Conversion from Original Image to Binary Image. We can Convert RGB to YCrCb Color Space and Segment Hand Region from Original Input Image. (a) RGB Image. (b) YCbCr Image. (c) Binary Image

Thus gestures can be found correctly if light intensity is varying.

##### 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 [12]. 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. Mathematically, Erosion is given by,

$$A \ominus B = \{x \mid (B) x \cap A c = \emptyset\}$$

Where A denotes input image and B denotes Structure elements. The Structure element is operated on the Image using a Sliding window and exact matches are marked. Figure 6 shows a graphical representation of the algorithm.

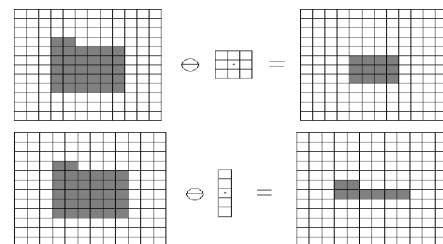


Figure 6: The Result of Image Erosion. The Structure Element Scans the Input Image. When the Structure Element Matches, the Central Pixel is Kept; when it Does Not Match, all Pixels are Discarded

Dilation is defined by,

$$A \oplus B = \{x \mid (B^{\wedge}) \cap A \neq \emptyset\} = \{x \mid [(B^{\wedge}) x \cap A] \subseteq A\}$$

Where A denotes the input image and B denotes the structure element. The same structure element is operated on the image and if the center pixel is matched, the whole area around that pixel is marked. Figure 7 shows the algorithm. Erosion and Dilation are changed by the shape of B. Thus, B should be selected in advance. Figure 8 is the result of this algorithm.

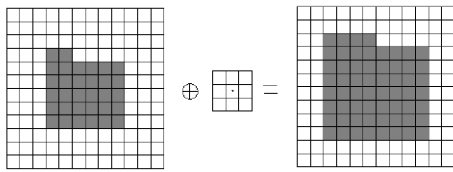


Figure 7: The Result of Image Dilation. The Structure Element Scans the Input Image. When the Center of the Structure Matches Any Pixels, the Bin of the Structure Element is Filled



Figure 8: Example Showing the Result of Image Morphology. Perform Erosion and Dilation on Binary Image. (a) Original Image. (b) The Result of Image Morphology

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 xi and yi 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.

To obtain the size of the hand, we will draw a circle increasing the radius of the circle from the center coordinate until the circle meets the first black pixel. When the algorithm finds the first black pixel then it returns to the current radius value. This algorithm assumes that when the circle meets the first black pixel, after drawing a larger and larger circle, then the length from the center is the radius of the back of the hand. Thus, the image segmentation is the most significant part because if some of the black pixels are made by shadows and illuminations near the center, then the tracking algorithm will meet earlier than the real background and the size of the hand region becomes smaller than the real hand. This problem breaks this system. Figure 9 shows the results.

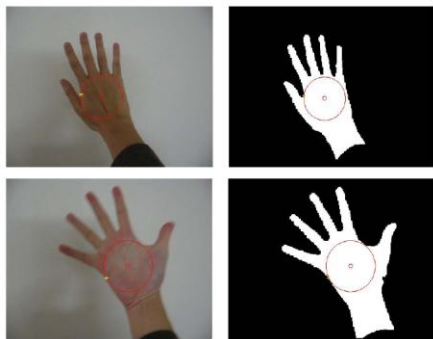


Figure 9: Example Showing the Result of the Computed Hand Size. Drawing a Larger and Larger Circle, we Obtained the Size of the Back of the Hand

E. Finding Finger Tip

The finger count is determined as in [13]. 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 [14].

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. A graph is used to count the number of transitions from white to black region. This number gives the number of active fingers.

From the number of active fingers the gesture made can be determined. If a value less than 0.7 is used the circle drawn encloses only palm region. If a value greater than 0.7 is used the circle doesn't intersect the thumb.

Finally, we can get one interesting point on each finger. Figure 10 shows the results.

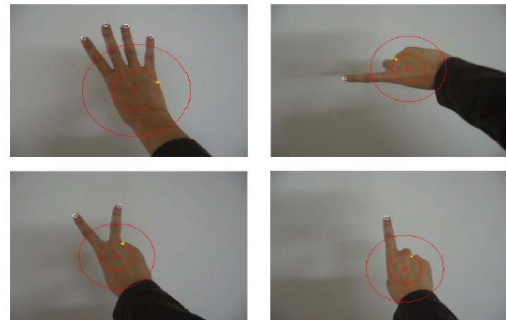


Figure 10: Example Showing the Result of Finger Tip Tracking Algorithm

V. CODE BLOCKS AND OPEN CV

The OMAP4 processor boards like Beagle board, Beagle board XM and Panda board come with full set of features like multiple OS support like Android, Windows Compact, Ubuntu, Cent OS, Meego and many more, we adopted an open source IDE called CODEBLOCKS which supports c, c++ programming and a library called Open CV in Ubuntu. Figure 11 shows screenshot of one of the simple example in Codeblocks using functions from OpenCV library. Figure 12 shows the result after segmentation and noise removal.

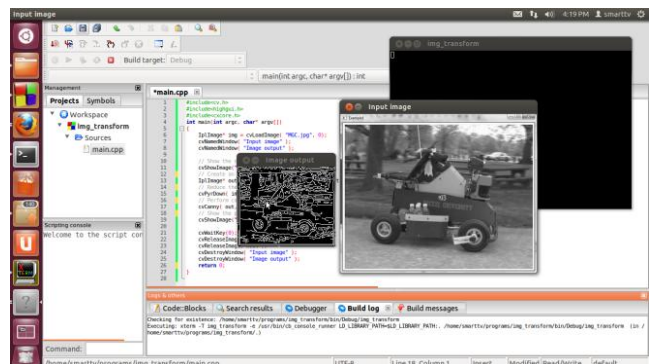


Figure 11: Code Blocks with Open CV Library

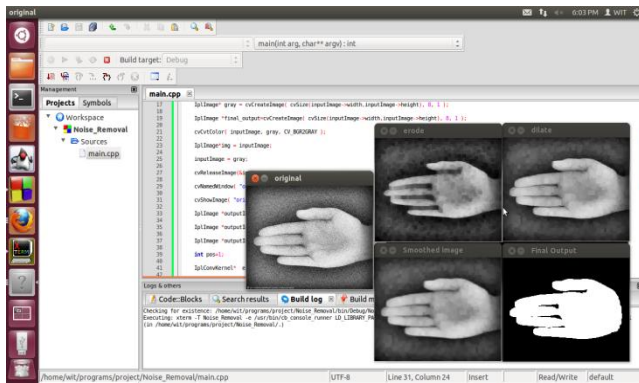


Figure 12: Result after Segmentation and Noise Removal

## VI. 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. Yet, input technologies seem to cause a major bottleneck in performing some of the tasks, under-utilizing the available resources and restricting the expressiveness of application use. Hand Gesture recognition comes to rescue here.

Technologies developed based on gesture are now really affordable and converged with familiar and popular technologies like TV, large screen. It's ubiquitous and non-intrusive as we can install a camera or remote with the TV. We will also conclude that a lot of applications can be created by using just the finger tips of a hand provided we will be able to distinguish between the fingers and will be able to track effectively.

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

## REFERENCES

- [1] Thad Starner, Alex Pentland, Visual Recognition of American Sign Language Using Hidden Markov Models, Massachusetts Institute of Technology
- [2] Lars Bretzner and Tony Lindeberg "Use Your Hand as a 3-D Mouse ...", Proc. 5th European Conference on Computer Vision (H. Burkhardt and B. Neumann, eds.), vol. 1406 of Lecture Notes in Computer Science, (Freiburg, Germany), pp. 141--157, Springer Verlag, Berlin, June 1998.
- [3] Matthew Turk and Mathias Kölsch, "Perceptual Interfaces", University of California, Santa Barbara UCSB Technical Report 2003-33
- [4] M Porta "Vision-based user interfaces: methods and applications", International Journal of Human-Computer Studies, 57:11, 27-73, 2002.
- [5] Afshin Sepehri, Yaser Yacoob, Larry S. Davis "Employing the Hand as an Interface Device", Journal of Multimedia, vol 1, number 2, pages 18-29
- [6] Henriksen, K. Sparring, J. Hornbaek, K. " Virtual trackballs revisited", IEEE Transactions on Visualization and Computer Graphics, Volume 10, Issue 2, paged 206-216, 2004
- [7] William Freeman, Craig Weissman, Television control by hand gestures, Mitsubishi Electric Research Lab, 1995
- [8] Do Jun-Hyeong, Jung Jin-Woo, Sung hoon Jung, Jang Hyoyoung, Bien Ze ungnam, Advanced soft remote control system using hand gesture, Mexican International Conference on Artificial Intelligence, 2006
- [9] K. Ouchi, N. Esaka, Y. Tamura, M. Hirahara, M. Doi, Magic Wand: an intuitive gesture remote control for home appliances, International Conference on Active Media Technology, 2005 (AMT 2005), 2005
- [10] Lars Bretzner, Ivan Laptev, Tony Lindeberg, Sören Lenman, Yngve Sundblad "A Prototype System for Computer Vision Based Human Computer Interaction", Technical report CVAP251, ISRN KTH NA/P--01/09--SE. Department of Numerical Analysis and Computer Science,

KTH (Royal Institute of Technology), SE-100 44 Stockholm, Sweden, April 23--25, 2001.

- [11] Hojoon Park Department of Computer Science Brown University, Providence, RI, USA, hojoon@cs.brown.edu
- [12] 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
- [13] Ram Rajesh J, Sudharshan R, Nagarjunan D, Aarthi R, "Remotely controlled PowerPoint presentation navigation using hand gestures", unpublished.
- [14] Asanterabi Malima, Erol Özgür, and Müjdat Çetin, "A fast algorithm for vision-based hand gesture recognition using for robot control", IEEE conference on Signal Processing and Communications Applications, Antalya, Turkey, 14th, April 2006.