

Air Gesture Library for Android using Camera

¹. Neeraj Phansalkar, ². Nabha Kumthekar, ³. Minal Mulay, ⁴. Sumit Khedkar, ⁵. G. R. Shinde

Department of Computer Engineering,
Smt. Kashibai Navale College of Engineering, University of Pune
Pune, India

Abstract— Android smartphones have completely revolutionized the mobile phone industry in the last five years. Gesture library of android is one of the advanced feature also one of the growing technologies for Smartphones. This air gesture library uses sensors for the gesture recognition. In this paper, we have proposed a system with front camera and Android Gesture Library to recognize different hand gestures. This system includes main steps: image capturing, image processing and gesture interpretation. The techniques used for Image Processing are skin colour detection, smoothing, hand extraction. For Image Processing we have used OpenCV library which has a collection of various Image Processing functions. This Air Gesture Library will lead to more intuitive user interfaces and not only make smartphones more accessible but user-friendly also for other devices using Android.

I. INTRODUCTION

In order to enable a better natural communication with virtual reality systems, hand gesture recognition appears as a suitable means. With the demands of time, the technology has changed from numeric keypads to QWERTY keypads then to touch screens and now it is no longer just a mobile phone with buttons and touch control only, but it can recognize gestures also. Hand gesture recognition provides easy, intelligent and natural way for human being to interact with Smartphones. It has vast range of applications in consumer electronics controlling systems like mobile phones. Controlling Android Devices via Air Gesture Recognition Library will be implemented for Android device users like Android phone, set top box etc. The aim of this system is to dissolve the interaction of human hand with their Android Devices by recognizing the hand gestures of users without touching the Device. Currently, few android phones have this feature for recognising hand gestures but, they use sensors which are often costly. In our paper, we are proposing a system which will have same air gesture functionality by using front camera.

The purpose of this system is to develop a library which can be utilized by any android developer to develop various android applications. By employing the facility of front camera efficiently and effectively user's gestures will be captured and based on proposed image processing techniques below the exact gesture will be recognized and depending on recognized gesture respective action will be taken. These gestures can be used to shuffle the songs in media player, to move the images in gallery and to scroll the browser window. Our library can be easily integrated with any application of android device having operating system version 4.0 and above (API Level 15 and above).

II. RELATED WORK

A growing class of smartphone applications are creating applications that run continuously, process data from sensors to determine the user's context (such as location) activity and optionally trigger certain actions when the right conditions occur. Currently there are very few android devices with device sensors to perform some gesture actions as required. Unfortunately many people cannot afford such devices because sensors are quite costly. There are some major drawbacks of current gesture recognition systems like high amount of battery usage, since continuous acquisition from sensors drains the battery, higher Cost of sensors and availability in very few android devices.

Many researchers have proposed various methods for dynamic hand gesture recognition using fingertip detection, but several limitations can be found in these approaches. In paper [1], 3D images method is used to recognize the hand gesture, but this process is complex and inefficient. Processing and execution time is very important and critical factor in real time applications [2]. RGB vector based method for skin detection in images [3] is proposed for the better efficiency. Many other approaches are dependent on specialized instruments and setup like the use of infrared camera [4], stereo camera [5], a fixed background [6][7] or use of markers on hand.

In our project, "Air Gesture Library for Android using Camera" wants to develop a library that enables Smartphones to use their front camera so as to detect the gesture that will be recognized and converted into the touch. AGL is a library which simplifies the rapid development of applications. It gives comfort to non-expert end users to easily operate their Smartphones. AGL provides an air gesture recognition library to automatically detect and recognize gestures. We would be bringing following major improvements by cancelling use of sensors to detect air gestures, control over battery usage addition of hover gesture providing selection operation, reducing cost of handsets, a simple library that clearly defines Triggers and corresponding Actions. So this library can be easily integrated within any android application. It's easy to use and cost effective.

III. PROPOSED AIR GESTURE LIBRARY

A. System Architecture

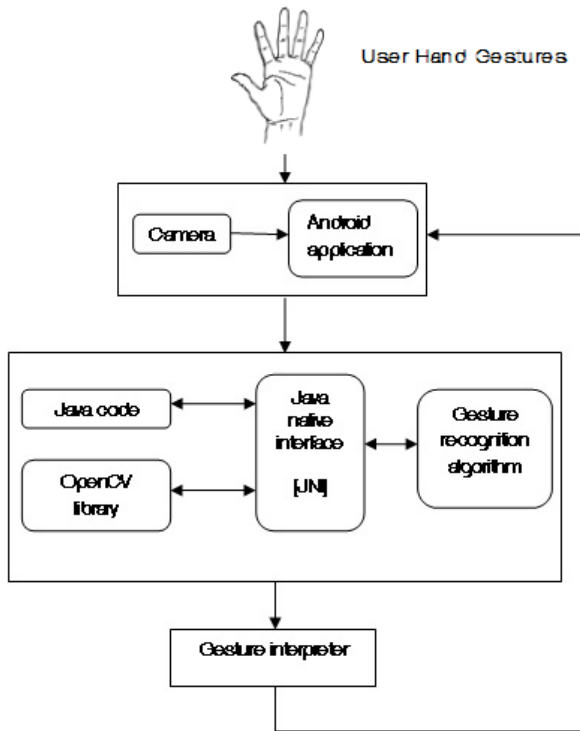


Figure 1. System Architecture

This system architecture includes:

- a) Capturing images (frames) continuously using front camera through android application and Passing those frames for image processing to main Air Gesture Library.
- b) Being main part of the system Air Gesture Library contains Java Native Interface, Image processing techniques and Gesture Recognition Algorithm.
 - i) JNI (Java Native Interface) acts as a communication medium between Java Code (accepts frames from android application) and OpenCV library (performs the task of Image Processing).
 - ii) On processing images we get the exact position of hand. This image processing technique includes following steps: Skin colour detection, Smoothing of image and hand extraction [8], [12].
 - iii) Gesture Recognition Algorithm takes the help of OpenCV library to find coordinates of hand and compares them. After successful comparison it returns a valid gesture type.
- c) Gesture Interpretation module sends the recognized gesture to respective android application.

Subsequently android application will perform necessary action.

B. Image Processing

Using front camera, we get RGB frames and to detect hand from those frames following phases are used:

a) Skin Colour Detection:

The main challenge in skin detection is to make the recognition robust to the large variations in appearance of skin that may occur, like in colour and shape, effects of occultation, intensity, location of light source, etc. In this very first phase, processing is required to detect the skin region from input frames. Typically the frames we get from camera are in RGB colour system. This colour system is not efficient for skin colour detection because of variations in lighting conditions. Hence, the result of this will not be consistent. So, to overcome this problem is another important task which is accomplished better by encoding the pixel's colour in each image frame using the HSV(Hue, Saturation, Value) nonlinear colour space representation, this is suitable for representing a range of colours. *Hue* defines the dominant colour (such as red, green, purple and yellow) of an area; *saturation* measures the colourfulness of an area in proportion to its brightness. The "intensity", "lightness" or "value" is related to the colour luminance.

b) Smoothing of Image:

Images are corrupted by random variations in intensity values called noise due to non-perfect camera acquisition or environmental conditions. Most smoothing methods are based on low pass filters. Gaussian filter is used to blur or smooth an image by Gaussian function. It is typically used to reduce image noise. Due to reduction in noise unwanted artifacts are removed so as to avoid unnecessary computations and increase in efficiency. Formula used by Gaussian Blur Filter can be stated as:

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

c) Hand Extraction:

While finding skin colour pixels for extraction of hand, there is a possibility of skin colour objects to remain in background. This main activity of hand extraction can be accomplished by using the concept of biggest BLOB (Binary Linked Object) detection [8]. BLOB can be classified using area, perimeter, aspect ratio, center of mass, best-fitting ellipse or rectangle. By using one of the above properties, we can get our objects of interest. After processing above phases, main task is to find out different objects involved in it and finding out the biggest

BLOB from given frame. As hand is closer to front camera, the area occupied by the hand will be the largest and hence it becomes the largest Binary Linked Object.

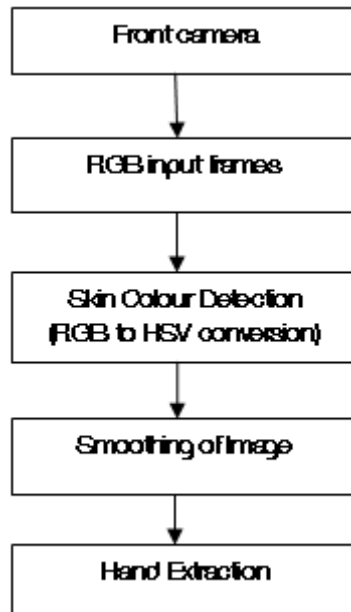


Figure 2. Image Processing Phases

IV. MATHEMATICAL MODEL

Following is the mathematical model for Android Air Gesture Recognition System. We are providing our mathematical model in terms of set theory.

- a) Let 'S' be the Air Gesture Recognition System. Hence it must contain some elements.

$$S = \{ \dots \}$$

Now, elements of the system are set of input, output and process elements and some other elements.

$$S = \{ I, O, P, \dots \}$$

- b) Identifying inputs and outputs of the system

$$I = \{ i \mid 'i' \text{ is video frame captured by front camera} \}$$

$$O = \{ o \mid 'o' \text{ is the set of output elements} \}$$

$$P = \{ 'p' \mid 'p' \text{ is the set of processes that are applying on the video frames} \}$$

- c) Identifying some processes.

$$P = \{ G, E, C, M \}$$

- i) G is the module for converting video frames from RGB format to HSV (Hue, Saturation, Value) format such that,

$$G = \{ G_{ip}, G_p, G_{op} \}$$

$$G_{ip} = \{ 'f_{in}' \mid 'f_{in}' \text{ is the valid frame coming from front camera} \}$$

$$G_{op} = \{ 'f_{out}' \mid 'f_{out}' \text{ is the output of this process} \}$$

$G_p =$ It is the function that accepts the RGB Frame coming from front camera and converts into the HSV format such that,

$$G_p(G_{in}) = G_{op}$$

- ii) E is the module used for smoothing the image.

$$E = \{ E_{in}, E_p, E_{op} \}$$

$$E_{in} = \{ 'k' \mid 'k' \text{ is the valid frame coming from module G} \}$$

$$E_{op} = \{ 'v' \mid 'v' \text{ is the blurred or smoothed frame} \}$$

$E_p =$ It is the function that processes the captured frame and then does some image processing to reduce the noise from frame.

$$E_p(E_{in}) = E_{op}$$

- iii) C is the module used for hand extraction and also extracts some parameters.

$$C = \{ C_{in}, C_p, C_{op} \}$$

$$C_{in} = \{ 'k' \mid 'k' \text{ is the valid frame coming from above module} \}$$

$$C_{op} = \{ 'v' \mid 'v' \text{ is the parameters of hand extracted from input frame} \}$$

$C_p =$ It is the function that processes frame coming from above module and detects exact shape of hand and find out some parameters.

$$C_p(C_{in}) = C_{op}$$

- iv) M is the module for mapping frame from air-gesture format to android gesture format such that,

$$M = \{ M_{ip}, M_p, M_{op} \}$$

$$M_{ip} = \{ 'm' \mid 'm' \text{ is the valid frame} \}$$

$$M_{op} = \{ 't' \mid 't' \text{ is the parameter extracted from input frame} \}$$

$M_p =$ This function maps the frame which is in air-gesture format to android gesture format such that,

$$M_p(M_{ip}) = M_{op}$$

- d) Identifying failure cases as 'F'. Failure occurs when there is matching of input to ideal one is less than 70%. Failure may also occur when frame contains non skin pixels detected as skin pixels at large amount.

- e) Identifying success case as 'e'.

Success occurs when input is matching with standard data more than 70%.

- f) Putting initial condition of the system as S_0 .

Hence, the overall air gesture system in terms of set theory will be:

$$S = \{I, O, P, F, e, S_0\}$$

V. CONCLUSION

In our paper we have described the library for Air Gesture Functionality in smartphones which can be easily utilized by Android developers for application development. For this purpose we have used Android Development Tool along with OpenCV library for image processing. This library will be available in the form of 'jar' file for easy import in any android application. Though the code is written in Java programming language, still there are various Android dependencies to use this library. In future, it has a wide scope in various home appliances like set top boxes, televisions, etc. (having Android Operating System only).

REFERENCES

- [1] Garg, P., Aggarwal, N. and Sofat, S. (2009). "Vision Based Hand Gesture Recognition", *World Academy of Science, Engineering and Technology*, vol. 49, 972-977.
- [2] Ozer, I.B., Lu, T. and Wolf, W. (2005). "Design of a Real Time Gesture Recognition System: High Performance through Algorithms and Software", *IEEE Signal Processing Magazine*, 57-64.
- [3] Aznaveh, M.M., Mirzaei, H., Roshan, E., Saraee, M., "A new colour based method for skin detection using RGB vector space", *Conference on Human System Interactions*, pp.932-935, 25-27 May 2008
- [4] Oka, K., Sato, Y., Koike, H. (May, 2002). "Real time Tracking of Multiple Fingertips and Gesture Recognition for Augmented Desk Interface Systems". *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR.02)*, pp.411-416, Washington, D.C., USA.
- [5] Ying H., Song, J., Renand, X., Wang, W. (2008). "Fingertip Detection and Tracking Using 2D and 3D Information. *Proceedings of the seventh World Congress on Intelligent Control and Automation*", pp.1149-1152, Chongqing, China.
- [6] Crowley, J. L., Berardand F., Coutaz, J. (1995). "Finger Tacking As an Input Device for Augmented Reality." *Proceedings of. International Workshop on Automatic Face and Gesture Recognition*, pp.195-200, Zurich, Switzerland.
- [7] Quek, F. K. H., Mysliwicz, T., Zhao, M. (1995). "Finger Mouse: A Free hand Pointing Computer Interface.", *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, pp.372-377, Zurich, Switzerland.
- [8] J.L. Raheja, K. Das and A. Chaudhury, "An Efficient Real Time Method of Fingertip Detection", *International Conference on Trends in Industrial Measurements and automation (TIMA)*, 2011, pp. 447-450.
- [9] T. Kapuscinski and M. Wysocki, "Hand Gesture Recognition for Man-Machine interaction", *Second Workshop on Robot Motion and Control*, October 18-20, 2001, pp. 91-96.
- [10] C. Yu, X. Wang, H. Huang, J. Shen and K. Wu, "Vision-Based Hand Gesture Recognition Using Combinational Features", *IEEE Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2010, pp. 543-546
- [11] J. Rekha, J. Bhattacharya and S. Majumder, "Shape, Texture and Local Movement Hand Gesture Features for Indian Sign Language Recognition", *IEEE*, 2011, pp. 30-35
- [12] L.K. Lee, S.Y. An, and S.Y. Oh, "Robust Fingertip Extraction with Improved Skin Color Segmentation for Finger Gesture Recognition in Human-Robot Interaction", *WCCI 2012 IEEE World Congress on Computational Intelligence*, June, 10-15, 2012, Brisbane, Australia.
- [13] E. Koh, J. Won and C. Bae, "On-premise Skin Color Modeing Method for Vision-based Hand Tracking", *The 13th IEEE International Symposium on Consumer Electronics (ISCE)*, 2009, pp 908-909
- [14] A. Malima, E. Ozgur and M. Cetin, "A Fast Algorithm for Vision - Based Hand Gesture Recognition for Robot Control", *IEEE*, 2006.