IJCSN - International Journal of Computer Science and Network, Volume 3, Issue 6, December 2014
ISSN (Online): 2277-5420 **www.IJCSN.org**
Impact Factor: 0.274

Follower Robot Using Android Programming

[1] Pratiksha C Dhande, [2] Prashant Bhople, [3] Tushar Dorage, [4] Nupur Patil, [5] Sarika Daundkar

[1]Assistant Professor, Department of Computer Engg.,
Savitribai Phule Pune University,
Pune, Maharashtra 411028, India

[2, 3, 4, 5] Department of Computer Engg., Savitribai Phule Pune University,
Pune, Maharashtra 411028, India

Abstract - In this article we described the attempt to build a robot able to locate and follow an human target moving in a domestic environment.[2] After a brief review of the state of the art in relative location technologies, we described our approach that aims to develop robots provided with simple and robust relative location technologies that do not require to structure the environment and on simple semi-reactive strategies that does not require the use of internal maps and the ability to self localize. More specifically, the approach is based on a control system able to display and integrate an exploration, obstacle avoidance, and target following behaviour and a relative location device based on an android programming with the help of image processing.

Keywords - *Android programming, Image processing, follower robot.*

1. Introduction

In this article we describe the research conducted in the attempt to develop a person follower mobile robot. Rather than structuring the environment by introducing android mobile or device that can allow the identification of the current position of the target person, we decided to provide the target person with the help of mobile capturing images with particular time interval and the robot with a sensor able to detect the current relative direction of the target person, we decided to provide the target person with a contrast colour object and the android device or mobile able to detect or capture contrast color object to detect the current relative direction of the target person.

1.1 Literature Survey

Existing system that provide the target person with a transmitter(that broadcast infrared and/or radio signals in the environment) and robot with sensor able to detect the current relative direction of the directed person.[2] Existing system that provide positioning information are often not a practical solution for mobile robotics. The proposed system provides a combination of android programming and concept of image processing for detection of movable object or target person. In this system android mobile can process video in real time and detect and follow specific object / shape i.e captures images of an contrast color movable object or target person at particular time interval and android app is used to apply several algorithms of image processing to detect the target person or object.

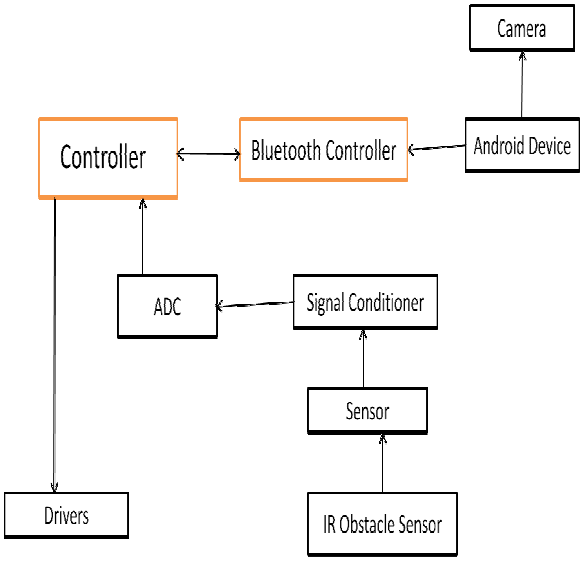


Figure 1: System Architecture

The system architecture can be divided into three parts:

1. Android programming: The proposed system starts with capturing images of target person and after that it performs several image processing

algorithms like RGB separation, Blur, Gray scale, Edge detection, Boundary detection, cropping, HSV etc.

2. Bluetooth connectivity: Bluetooth device sends signal towards the controller to detect current position of target person.
3. Serial Communication: Serial communication gives direction to robot according to signals which sends to the controller. Serial communication controls stepper motors with the help of driver IC.

2. Algorithms

The main steps of the algorithm used in application:

An RGB image is capture by mobile phone camera then image is converted into greyscale.
Further, a key point detection procedure is performed.
The key descriptors are calculated on the basis of local image gradient magnitude and orientations.
Key point descriptor is based on gradient magnitude captured for 16 or 4 pixel adjacent to key point. These values are used to form the so called edge detection histogram.
A gradient magnitude 'm' and a gradient orientation 'ɵ' of each pixel is given by formula

$$m(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((I(x,y+1) - I(x,y-1))/(I(x+1,y) - I(x-1,y)))$$

Where, I(x,y): Brightness value of pixel at x,y coordinate.[4]

Object detection on android involves the following steps:

1. A binary image is produced using thresholding method.
2. Morphological opening and closing filters are used in sequence for object detection.
3. Contour based learning technique is used for drawing the contours for the objects detected and extracting them for further analysis.

In this proposed system we will use algorithms RGB separation, Blur, Grayscale, Edge detection, Thresholding, Boundary detection, cropping, HSV.
Different Image Types
• COLOR – GRAYSCALE – THRESHOLD IMAGE

Original **Grayscale** **Threshold**

Figure 2 COLOR-GRAYSCALE-THRESHOLD IMAGE.

2.1 RGB Separation

The RGB colour model is an additive colour model in which red, green, and blue colour is added together in various ways to reproduce a broad array of colors.

Color images are composed of three different channels viz. Red, Green and Blue.

• One can also imagine three overlapped 8-bit images to compose a final 24-bit color image.
• In computing, the component values are often stored as integer numbers in the range 0 to 255. These may be represented as either decimal or hexadecimal number.

RGB Color Example
• Sample PIXEL value in HEX = 0EDEB5
• In programming the hex numbers are represented as 0x0EDEB5. 0x prefix is for hex notation.
• Then individual colour channels:
• 0E (red) - DE (green)- B5 (blue)
• 00001110 – 11011110 – 10110101
• Actual Colour Composed Will Be:
Traverse Through Entire Image Extract 8-bit R, G and B values from 24-bit Colour Value

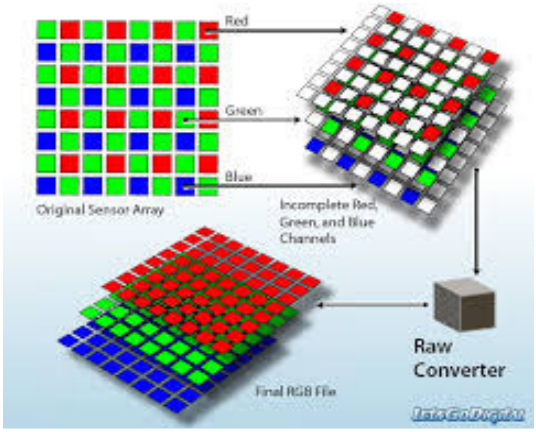


Fig. 3 RGB Separation

2.2 RGB to Greyscale Conversion

In a (8-bit) greyscale image each picture element has an assigned intensity that ranges from 0 to 255.

• A grey scale image is different from black and white image since a greyscale image also includes shades of grey apart from pure black and pure white color.
• Greyscale images are usually required for image processing.

• Steps / Algorithm
– Traverse through entire input image array.

– Read individual pixel colour value (24-bit).
– Split the colour value into individual R, G and B 8-bit values.
– Calculate the greyscale component (8-bit) for Given R, G and B pixels using a conversion formula.
– Compose a 24-bit pixel value from 8-bit greyscale value.
– Store the new value at same location in output image.

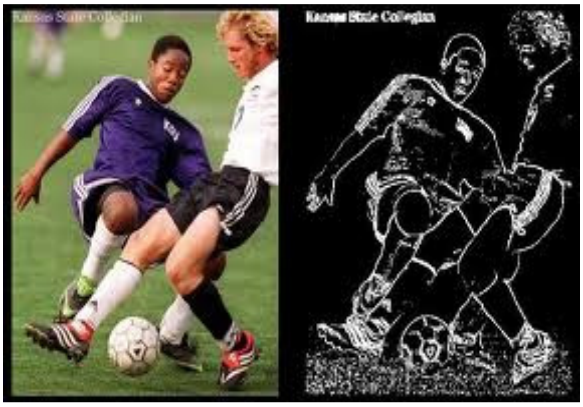
Original Image **Grayscale Image**

Fig. 4 RGB to Grayscale Conversion

2.3 Edge Detection Algorithms

Edge detection algorithms of computer vision techniques are used to detect the edges and tag objects. All the edges in the image were detected. But it is not useful for identifying individual objects in image. Identifying those objects from the detected edges is difficult and long process.

In the present approach it has been attempted to convert the colour image to binary image using thresholding function.



Original Image **Edge detection image**

Fig. 5 Edge Detection

2.4 Thresholding Algorithm

• Thresholding is the simplest method of image segmentation.
• From a greyscale image, thresholding can be used to create binary images i.e. image with only black or white colors.
• It is usually used for feature extraction where required features of image are converted to white and everything else to black. (Or vice-versa)

The following figure shows first conversion of original color image into greyscale image and after conversion of 8 bit greyscale image it again converted into threshold image.

Original Image **Grayscale Image**

RGB to Grayscale Conversion

Greyscale Image **Threshold Image**

Grayscale to Threshold Conversion

Fig. 6 Threshold Image Conversion

Thresholding

• Steps / Algorithm

– Traverse through entire input image array.
– Read individual pixel color value (24-bit) and convert it into greyscale.
– Calculate the binary output pixel value (black or white) based on current threshold.
– Store the new value at same location in output image.

2.5 HSV MODEL

H (hue) - Specify the position of pure color on wheel.
S (Saturation)-Describe the how white the color is. E.g. pure red is fully saturated; tints of red have saturations less than 1.

V (Value)-called as 'lightness of color'. Describe intensity of color . Can be described as brightness in the color.

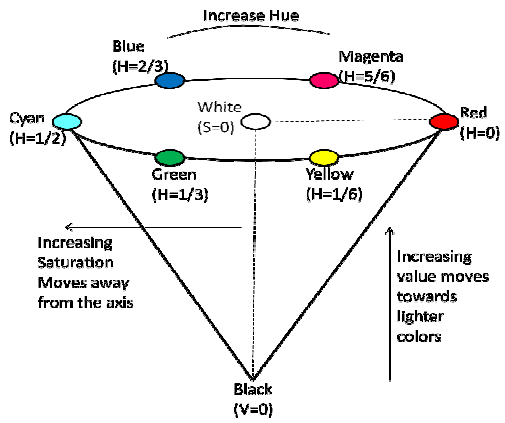


Fig. 6 HSV Model

Advantages of HSV over RGB

• Strong model than RGB because it offers a more intuitive representation of the relationship between colors.
• HSV selects more specific color.
• In HSV model value of 'H' and 'S' remain constant if the value of 'V' changes, but value of RGB changes with the change in 'V'.

3. Mathematical Model

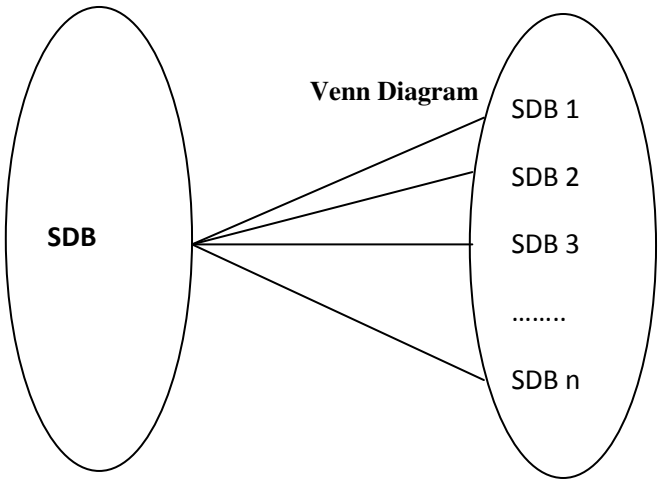
Set Theory:

Let s (be a main set of) ≡ {SDB, LDB, C, A, S}
Where, SDB is the copy of the server database. This database is responsible for storing user information.
LDB is a set of local database that a user owns. It consists of data tables having data items related to local environment.
C is a set of all clients using the follower robot app. And $(c_1, c_2, c_3, \ldots\ldots\ldots c_n) \in C$.
A is a set of algorithms applied on the input data to get image processing results.
S is the server component of the system. The server is responsible for registering, authenticating and providing associations to the end user.
Functionalities:
SDB' = RegisterUser(image_storage);
process 1= RGB_generation(input_image);
process2 = Blur(process1);
process3= Grayscale (process2);
process4= EdgeDetection(process3);
process5= Thresholding(process4);
process6= Blob(process5);
process7= Cropping(process6);
process8= HSV(process7);
process9= Histogram(process8);
process10= Normalization(process9);
process11= Registration(process10);
process12= Recognition(process11);
U = Authenticate User;
UPLOAD(image);
S= Scanned image and send signal to controller;
Results = Decode(S);

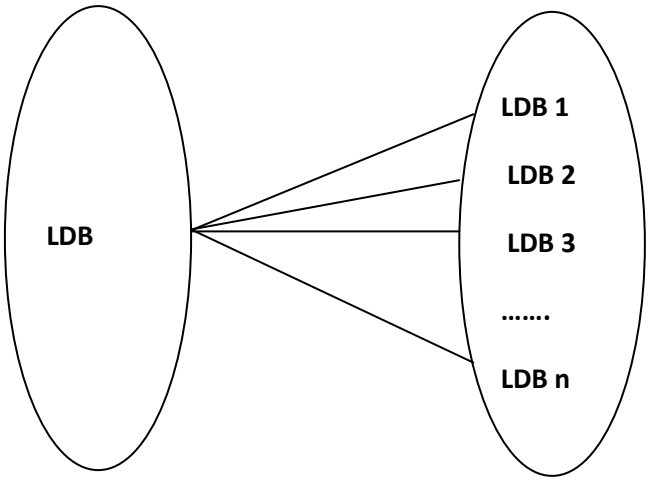
3.1 Activities

- **Activity 1**
  SDB is the copy of the server database. This database is responsible for storing user information.

- **Activity2**
  LDB is a set of local database that a user owns. It consists of data tables having data items related to local environment.



As described in Activity1 following Venn diagram is



As described in Activity 2 following Venn diagram is

3.2 P Complete Problem

P- Complete problem are solvable in polynomial in time. As an image and data have finite size and in brightness in color robot recognizes the color and follow the movable object or target person.

Hence here given input has finite appropriate solvable output.

Here our problem in P- completes.

3.3 Morphism

Since there can be n number of clients requesting to same system node for some method to execute ; let there be group of clients referring to system node.

For example, f(C)→{C1, C2, C3… Cn} ϵ C. This can be morphism of clients and clients can be independently drawn leading to concurrently approaching to same node.
In this system per client one allow node will execute the desired request hence for n number of client n number of requests will be delivered leading to $O(n^2)$. But with morphs it is always O(1) hence efficient.
Morphism with Concurrency leading to efficiency.

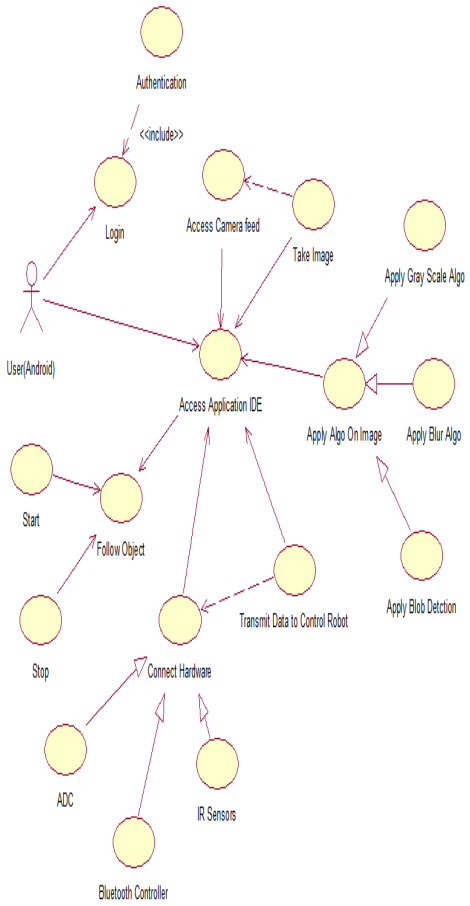
3.4 Overloading

If (T<=CNn(T)) then apply DTW.

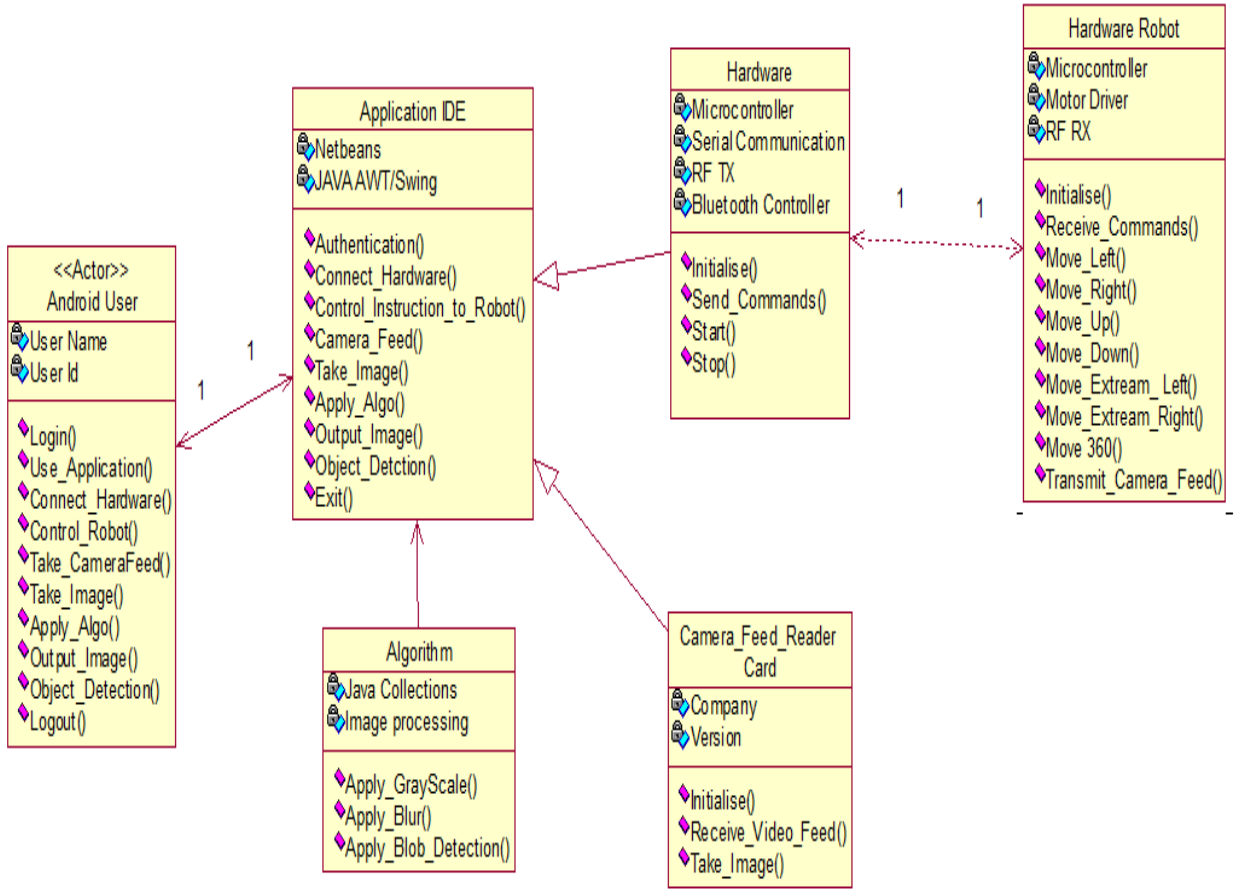
Overloading on (T, CNn(T)) will reduce If-then-else conditions.

4. UML Diagrams

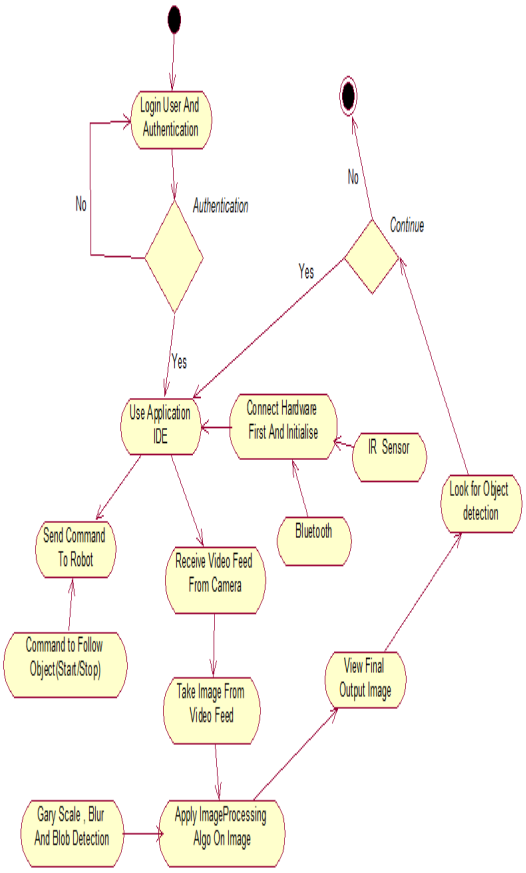
4.1 USE CASE Diagram



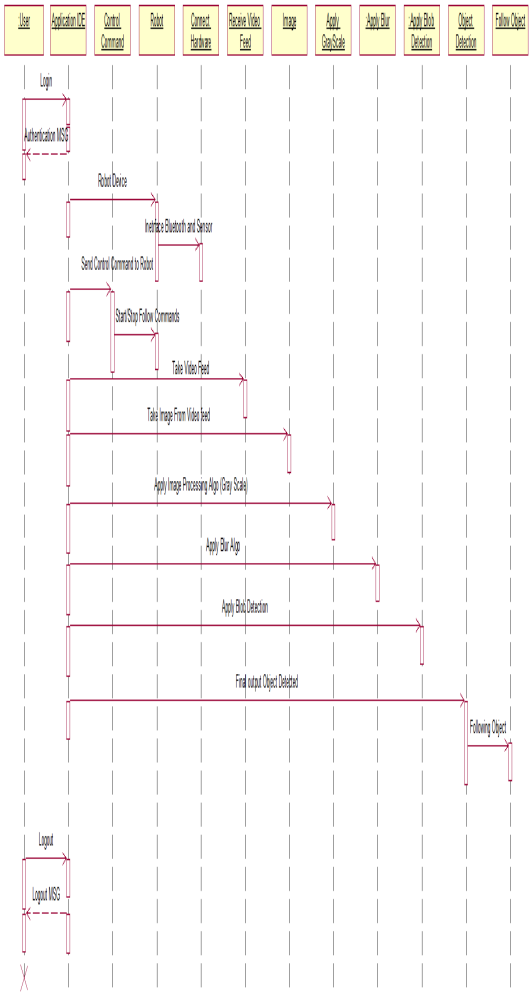
4.2 CLASS Diagram



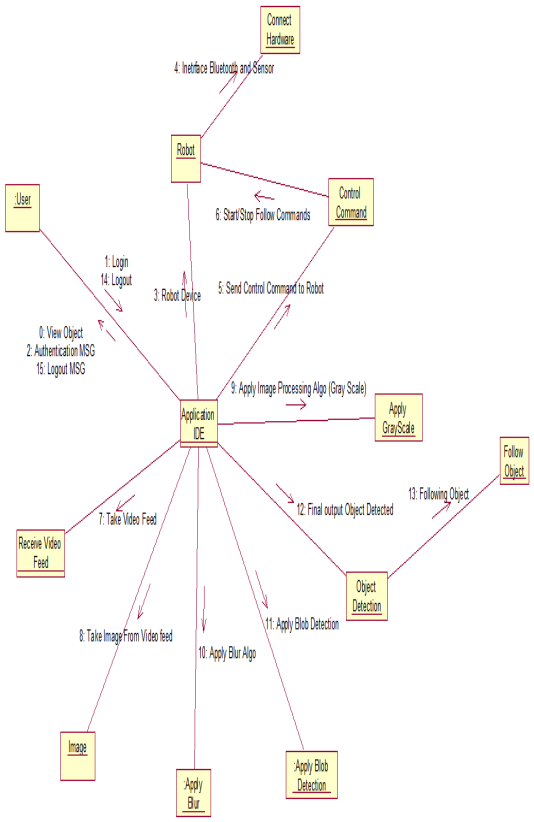
4.3 Activity Diagram



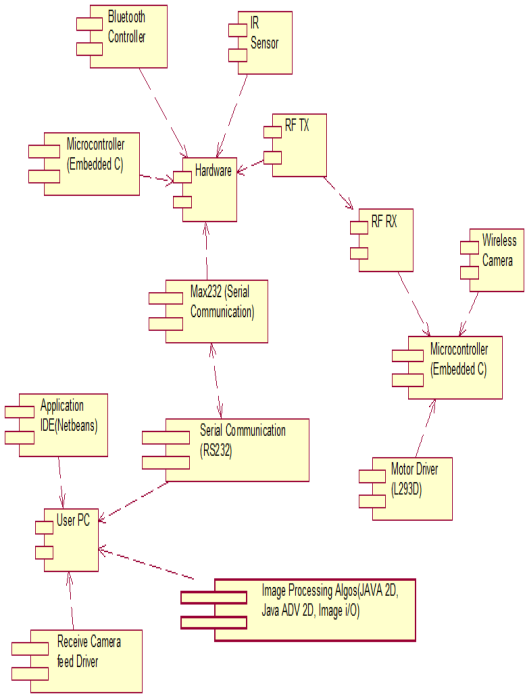
4.4 Sequence Diagram



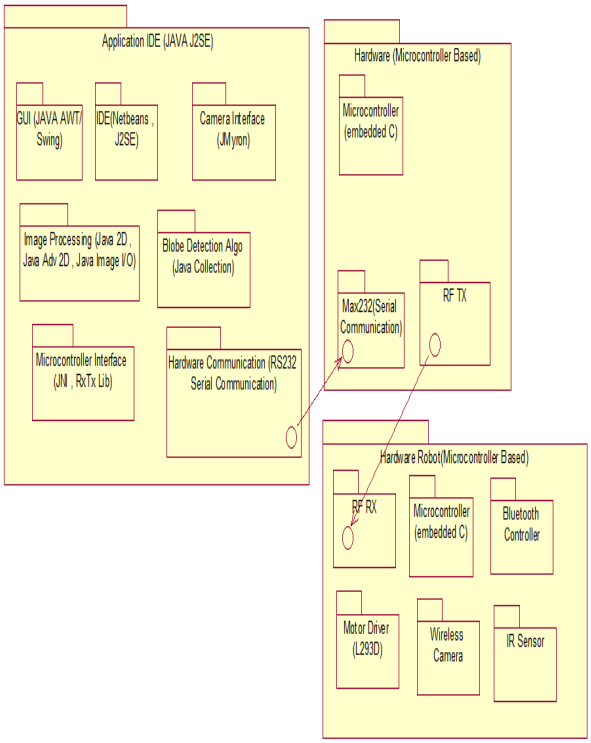
4.5 Collaboration Diagram



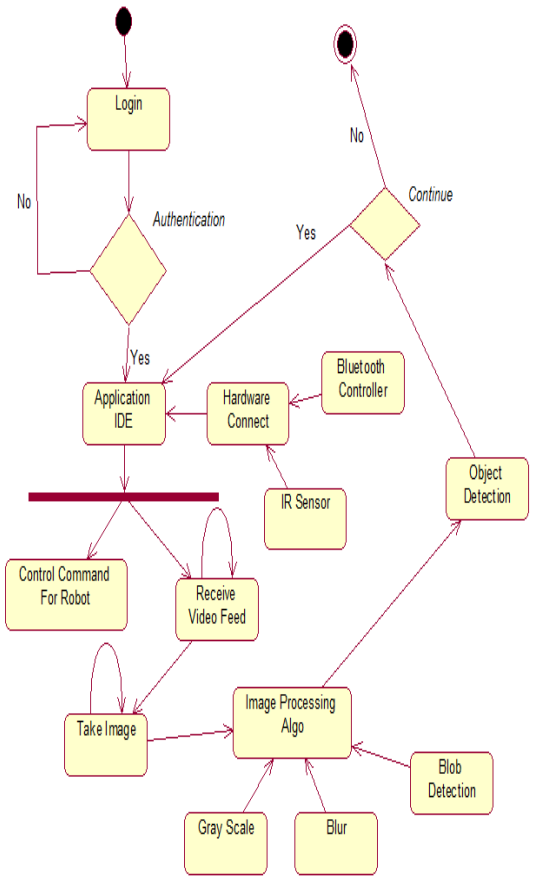
4.6 Component Diagram



4.7 Package Diagram



4.8 State-Chart Diagram



5. Conclusion

Object detection is achieved using android development tool and Java language and implemented on android device by using image processing algorithms and android programming. Larger objects got detected and indicated by marking their boundary.

Reference

[1] 2014 Fifth International Conference on Signals and Image Processing 978-0-7695-5100-5/13 $31.00 © 2013 Crown Copyright DOI 10.1109/ICSIP.2014.69 An Approach for Object Detection in Android Device.

[2] Toward a Person-Follower Robot Copyright © 2003, The ROBOCARE Project Funded by MIUR L. 449/97. All rights reserved.

[3] 2011 First ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial Engineering *Object Detection using FAST Corner Detector based on Smartphone Platforms* 978-0-7695-4417-5/11 $26.00 © 2011 IEEE DOI 10.1109/CNSI.2011.60.

[4] Object recognition in a mobile phone application for visually impaired users978-1-4673-5637-4/13/$31.00 ©2013 IEEE.

Pratiksha Dhande received the masters in Computer Science and Engineering(M Tech).

Prashant Bhople received the Diploma degree in computer engineering from Maharashtra State Board of Technical Education, Pune, Maharashtra in 2008, Where he is currently persuing Bacholar of degree in Computer Engineering.