

ANDROID BASED REMOTE TENNIS BALL COLLECTOR

CS308: ERTS Lab
Group 14

Team Members

Hasan Kumar Reddy A (09005065)

Bhanu Prakash V (09005050)

Vinod Reddy G (09005071)

Avinash Theetla (09005056)

Problem Statement

- Aim : To control the Fire Bird V motion using an android mobile
- Utility : Tennis ball collection
- Modes : Autonomous mode and Manual Mode
- Notes :
 - Image Processing using Open-CV on android mobile
 - Remote Live Video Transmission via Wi-Fi Camera
 - Mobile to Bot communication via Bluetooth

Description

Our Fire Bird V functions as a tennis ball collector. It sends the video stream of environment in front of it via an IP Camera fixed in front of it. Android mobile downloads the video from the IP Address of camera and processes it for the presence of tennis ball. Then it sends appropriate signal to the bot via Bluetooth communication module.

Motivation

Setting up a PC is a cumbersome task, especially in times of emergency. Hence we switched to android mobile for image processing and sending the signals.

Requirements Aimed

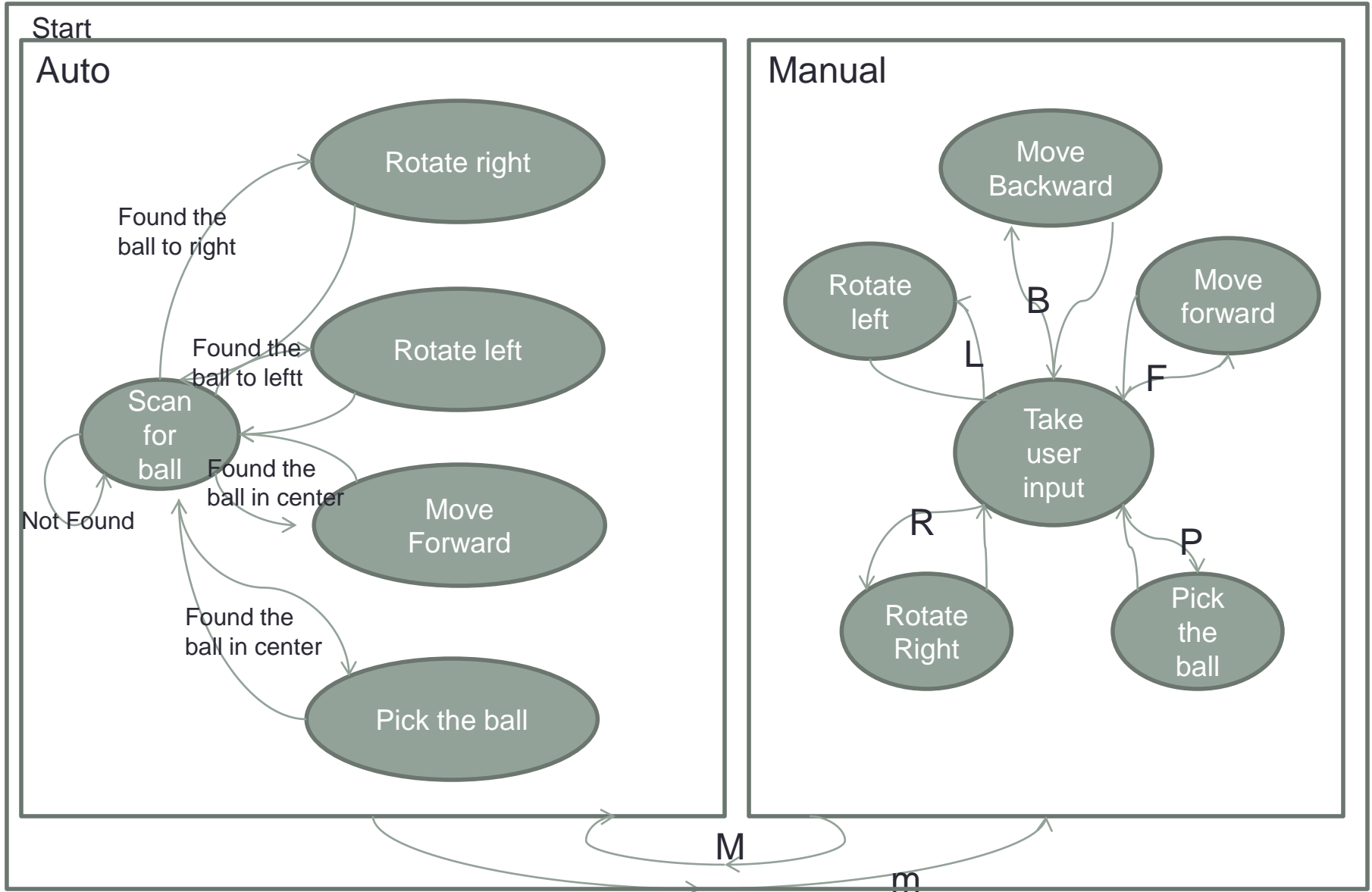
S.No.	Aim	Status
1.	Remote Video Transmission via Wi-Fi (Hardware)	completed
2	Android Application (Software)	completed
3.	Image Processing on Android Mobile (Software)	completed
4.	Bluetooth signal Transmission (h/w + s/w)	completed
5.	Ball Picker (Arm) Construction	done
6.	Firebird application code	completed
7.	Completely wireless	done

Project Plan

S.No.	Milestone	Alloted to	Completed On
1.	Project Presentation	Vinod, Hasan	7 th March
2.	Identifying modules and allotment	All	16 th March
3.	Setting up the Wi-Fi Camera	Hasan	29 th March
4.	Received our order for Bluetooth module		3 rd April
5.	Android Application	Hasan	7 th April
6.	Image Processing Code	Vinod	9 th April
7.	Firebird Application Logic	Bhanu	8 th April
8.	Fixing Arm on Bot	Avinash	9 th April
9.	Bluetooth Transmission	Avinash, Hasan	10 th April
10.	Bluetooth Receiving	Bhanu, Vinod	10 th April
11.	Project Integration	All	11 th April

- Weekly team-meet and work review - Not strictly followed.

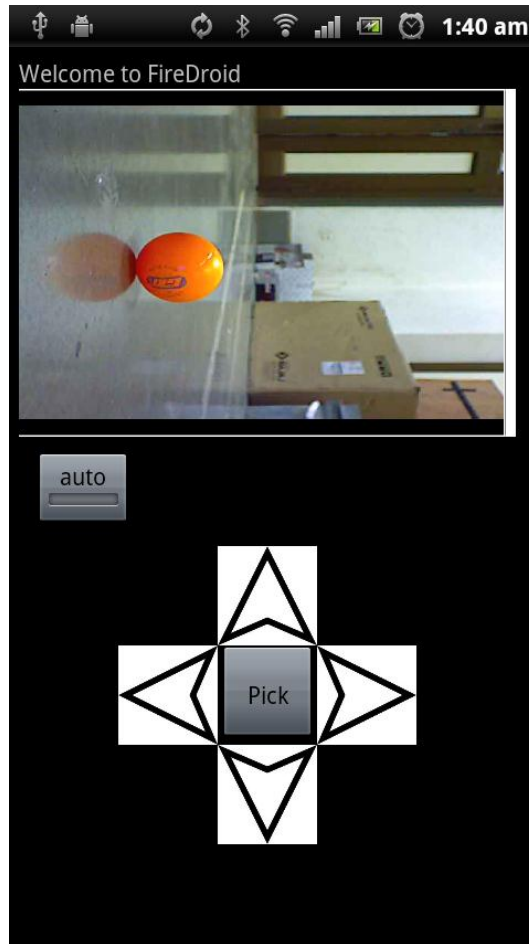
Finite State Machine



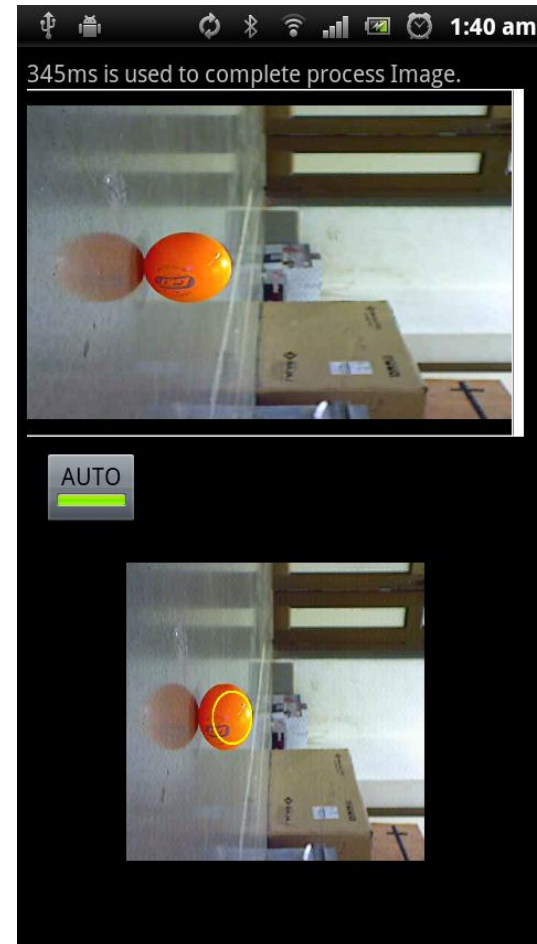
Tasks, Challenges & Innovation

- Video Transmission
 - Android doesn't recognize the usual Ad-hoc network setup by PC
 - Tried
 - Rooting mobile, using various apps.
 - Setting up a WLAN network bridge in PC - Worked
- Android Application
 - Loaded the video in a webview
 - Default view from IP Camera shows four streams
 - Ran Javascript on top of it to display just the video required.
 - Initially loaded a snapshot and regularly updated it at fixed intervals
 - Java timer made the app non-responsive sometimes

Tasks, Challenges & Innovation



Manual Mode



Auto mode

Tasks, Challenges & Innovation

- Open CV
 - Installation on windows – a challenge by itself
 - Tried many work arounds, online tutorials
 - Finally could get it working – links in report
 - Linking the OpenCV C++ code with Android Java
 - Android jni – relatively simpler
 - Usual approach using cvHoughCircles – shape detection – slow
 - Devised a new approach
 - Pixel by pixel analysis

Tasks, Challenges & Innovation

- Hardware

- Camera mounting

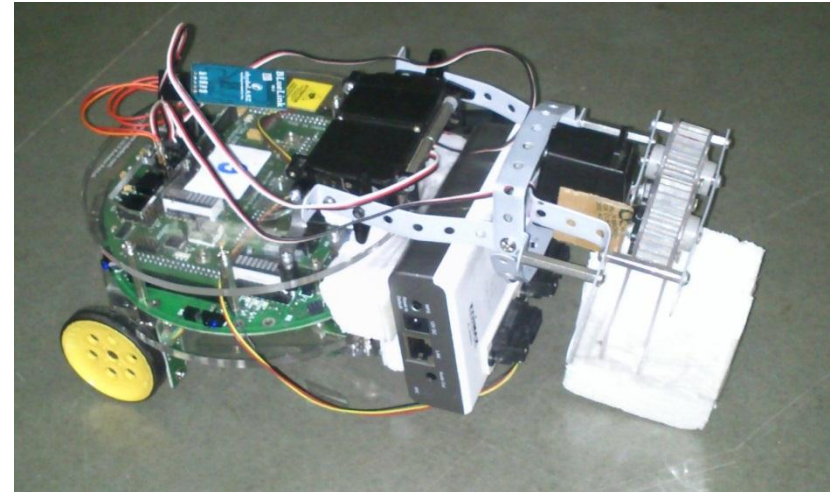
- Front
 - will block the sharp sensors
 - will block grippers movement
 - Top
 - Gripper arm in its path

- Sensor

- Single sensor – missed the ball many times

- Solution

- Increase the arm's length
 - Place 2 sensors ON the Camera
 - Fix the Camera in front of bot



Tasks, Challenges & Innovation

- Bluetooth Setup
 - Identification & Procurement
 - Help from Piyush, Lab Asst.
 - Programming
- Firebird Logic Implementation
 - Identifying the correct sensor(s) reading under which the gripper arm should be signaled to pick the ball
 - Understanding (and fixing the gripper arm) the behaviour of gripper arm for various values servo motor

Assumptions & Limitations

- Only orange colored objects in view are TT balls
- Only one ball in the bots view
 - Though our openCV code can distinguish and identify one of the balls, bot might get confused in few cases
- Sensor value might change with hardware
- Hardware
 - Android :
 - armv7 processor is recommended, because of have a floating point preprocessor (Open CV).
 - Touch enabled – by design
 - Bluetooth and Wi-Fi capability – most smart phones have it
- Basket not implemented
- Build only for android $v \geq 2.3$
 - Bug in 2.2 regarding BitmapFactory

Test Criteria and Delays involved

- Introduced a delay of 1s between any two consecutive communication.
 - Android (Sony MT11i) takes about 0.5 – 1s to download, process and transmit the signal
 - Much less resources when compared to PC
 - Share your PCs network with the WLAN setup,
 - Otherwise, the download seemed to be very slow (20 s), apparently no reason which we could find
- Once the ball is picked in auto mode, cannot pick any more balls until reset.
 - By design
- Drawbacks
 - Android application seemed to crash when many other applications are running simultaneously.
 - May be using more runtime memory
 - Close all other applications before you start the app

Re-usability

- Have provided brief documentation and references to *Open CV in android* setup and usage
 - Could save a lot of effort
 - Also included the jni resources – cross compilation and linking
- Modular nature
 - Image Downloading
 - Image Processing
 - Communication
 - Application

codes are all separated via different classes.
- In code documentation standards
 - Followed the usual Java commenting style (eclipse friendly)

Future Enhancement

- Replace the Bluetooth communication with Wi-Fi shield
 - <http://www.instructables.com/id/How-to-build-an-Arduino-WiFi-4x4-with-Android-Cont/>
 - http://asynclabs.com/store?page=shop.product_details&product_id=26&vmcchk=1
 - Will make the whole application really remote
- Use an existing or write an Open CV wrapper for android
 - <http://code.google.com/p/javacv/>
- Improve the UI control – use joystick
 - Good project
 - <http://www.instructables.com/id/JabberBot-The-Arduino-robot-with-an-ATMega-brain-/>