CS-308-2014 Final Report

Intruder Detection

Mavericks

TH-08

Raghav Sagar, 100050001

Anil Kumar, 100050013

Vinod Talapa, 100050014

­Ashish Lalawat, 100050017

Table of Contents

[1. Introduction 3](#__RefHeading__27_1832536793)

[2. Problem Statement 3](#__RefHeading__29_1832536793)

[3. Requirements 3](#__RefHeading__31_1832536793)

[3.1 Functional Requirements 3](#__RefHeading__33_1832536793)

[3.2 Non-Functional Requirements 3](#__RefHeading__35_1832536793)

[3.3 Harwdare Requirements 4](#__RefHeading__37_1832536793)

[3.4 Software Requirements 4](#__RefHeading__39_1832536793)

[4. System Design 4](#__RefHeading__41_1832536793)

4.1 Overall design …………………………………………………………………………………………………………………4

4.2 System setup……………………………………………………………………………………………………………………5

4.3 State Chart ………………………………………………………………………………………………………………………6

[5. Working of the System and Test results……………………………………………………………………………….](#__RefHeading__43_1832536793)...7

5.1 How the system works…………………………………………………………………………………………………….7

5.2 Testing Functional requirements …………………………………………………………………………………….7

[6. Discussion of System……………………………………………………………………………………………………………..](#__RefHeading__45_1832536793)8

[7. Future Work………………………………………………………………………………………………………………………….](#__RefHeading__47_1832536793).8

[8. Conclusions…………………………………………………………………………………………………………………………..](#__RefHeading__49_1832536793)9

[9. References……………………………………………………………………………………………………………………………](#__RefHeading__51_1832536793)9

## 1. Introduction

## We have built an "Intruder Detection System". The motivation for the project is detection of unknown objects using a fixed surveillance camera in our arena. This is followed by providing live video feed of this unknown object, by our patrol bot. This is useful in detection of animal intruders in restricted areas (for example Greenhouse).

## 2. Problem Statement

## First step is detection of intruder using image processing on the video feed by our surveillance camera. After intruder detection, we have find its location so as to automatically navigate our patrol bot (Firebird V). After patrol reaches the intruder, the patrol bot will give the live video feed of the intruder. For manual control of the patrol bot, we have built a web application which gives the live video feed from the patrol bot with manual navigation using arrows keys.

## 

## 3. Requirements

### 3.1 Functional Requirements

* **Intruder Detection:** Detect presence of unknown intruder using surveillance
* **Locating Target:** Locating the position of the intruder in our arena
* **Navigate Patrol Bot:** Based on the location found for the intruder, the patrol bot is automatically navigated to this location.
* **Live Video feed:** Patrol Bot has a mobile camera attached in front of it, which provides live feed of the intruder.

### 3.2 Non-Functional Requirements

* **Digital Image Processing:** Image processing on the live feed from the surveillance camera is used to detect intruder
* **Camera Calibration**: Initially at least 4 points with known world coordinates are used for camera calibration. After this step, we can convert pixel coordinates to world coordinates
* **Patrol Bot detection:** We detect the patrol bot using a unique colour. This is needed to differentiate between bot and an intruder
* **Automatic Navigation:** Worlds coordinates of patrol bot and intruder are used for automatic navigation of bot to intruder location

### 3.3 Harwdare Requirements

* **Computer:** PC/Laptop with at least 1 GB RAM, USB connectivity, WiFi card and Windows OS
* **Patrol Bot:** Firebird V Bot with Xbee module
* **Surveillance Camera:** USB Webcam with at least 2 Megepixel camera and stand for mounting
* **Mobile Camera:** Mobile with at least 2 Megepixel camera and Andriod OS

### 3.4 Software Requirements

* **Driver for USB Webcam:** Driver for surveillance USB camera
* **Matlab:** Code related to image processing and patrol bot automatic navigation is written in Matlab[1]
* **Wampserver**[2]**:** Web application written in PHP, runs on an Apache server simulated on the local machine
* **Python:** Python[3] with PySerial[4] module for serial communication with Patrol Bot using Xbee module
* **Keil µVision:** ARM c/c++ compiler for Bot movement code.

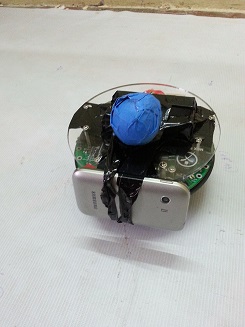
## 4. System Design

**Overall Design:**

Components: Arena, Surveillance camera, FB5 bot with mobile camera installed on it, Xbee Module.

Arena is monitored by an inclined surveillance camera. Arena have a patrol bot looking for intruder. Patrol bot is controlled by Intruder detection program running on Main PC via Xbee module. And the system also consists of web based application for getting live video feed using mobile camera installed on patrol bot.

**Snapshots of Mechanical Requirements:**

FB5 with mobile camera USB camera(surveillance) Xbee Modul

**System setup:**

­

Surveillance Camera

FB5 patrol bot

Calibration points

5

4

3

2

1

X-bee Module

Laptop

Arena

Intruder

­

­

**State chart**

**Initial Setup**

**Execution**

Take Base Image without bot and intruder

Camera Calibration

Set up an arena with surveillance camera

Deploy the bot movement embedded c code on bot

Introduce patrol bot in the areana

Connect XBee to the pc to create a serial communication port

Run the main.m file on the pc with XBee attached

Introduce intruder in the arena

Bot reaches it after sometime and stops

Stop the matlab script and run the web appapplication

Control the bot’s using keyboard to get the proper live video feed

## 

## 5. Working of the System and Test results

**How the System works:**

* **Camera calibration:**

We used Michel Chasles equation

which relates the 2D image pixel coordinates(x,y) and 2D coordinates in the World Coordinates (x’, y’) using 8 coefficients. So, we require to know the 2D coordinates of at least 4 points on the arena to find the 8 coefficients. For robust estimation, we added more such points to get more accurate results.

* **Intruder Detection:** Firstly, we detected the patrol bot in our arena so that is would not be considered as an intruder. The bot is detected using blue component. We took the image difference between the base image and the current image. If there is motion, there will be non-zero difference at those pixels. Also, we consider the connected components of pixels with non-zero difference. All the components with area greater than the threshold are considered. And then the centroid of the component with the largest area is considered as the intruder location.
* **Locating target:** After we got the intruder location in the image, we compute the corresponding world coordinates using the equation used in camera calibration.
* **Automatically navigating the bot to the target:** Up to this point, the world coordinates of the bot and intruder are known. We move the bot forward by small distance, so that we get the direction of the bot movement. Using the direction of the bot movement, and the direction between the bot and intruder, we get the angle required to rotate the bot to head it towards the intruder. Then, bot is moved forward in the rotated direction. The above method is repeated until the distance between the bot and the intruder is smaller than the threshold.
* **Manually navigating the bot:** We used Xbee module and made a Web application.
* **Live Video Feed:** The mobile camera is connected to the user device through wifi.

**Testing Functional requirements:**

* **Intruder Detection:** We introduced intruders of different types with different size and colour (except blue), in the testing arena and then tested whether our image processing algorithm detected it properly in each test case.
* **Locating Target:** After the intruder detection is done, we found manually the position of the intruder in world coordinates, and then compared it with the coordinates returned by transforming the pixel coordinates into the world coordinates.
* **Navigate Patrol Bot:** On detecting the intruder, we checked whether the PB reaches the original intruder site or somewhere else.
* **Live Video feed:** We checked whether the user is getting the proper view of the video recorded by the camera.

## 6. Discussion of System

a) What all components of your project worked as per plan?

* Intruder Detection
* Locating Target (Intruder)
* Automatic Navigation of Patrol Bot
* Manual Navigation of Patrol Bot using web application
* Live Video feed

b) What we added more than discussed in SRS?

* Web application for manual navigation (instead of android application)

c) Changes made in plan from SRS:

* Web application: We initially planned on manual navigation using Bluetooth module of Firebird V. But Bluetooth connection was very unreliable with frequent disconnections. Therefore we used Xbee module instead and therefore made a Web application instead of an android application

## 7. Future Work

* **Feature matching:** If better surveillance camera with higher resolution is used, during image processing, we can use feature matching to detect patrol bot
* **Adding obstacles:** The arena can include obstacles, and during automatic navigation, patrol bot should navigate around them
* **Fire detection:** Temperature sensors can be used to detect fires. Feature matching for fires can also be used

## 8. Conclusions

* Successfully built an ‘Intruder Detection System’ using image processing
* Location detection of intruder followed by automatic navigation of patrol bot
* Live video feed of intruder on a web application with manual navigation controls for patrol bot

## 9. References

[1] Matlab: <http://www.mathworks.in/products/matlab/>

[2] Wampserver: <http://www.wampserver.com/en/>

[3] Python: <http://www.python.org/download/>

[4] PySerial Module: <http://www.pyserial.sourceforge.net/>

[5] Camera Calibration: http://goo.gl/2cK0bw