University of Brighton

School of Computing engineering and mathematics

BSC COMPUTER SCHIENCE (2018-19)

MODULE: CI301, The individual project

Interim Planning and Investigation Report

Project: Raspberry Pi Surveillance system

STUDENT:
BSC (HONS) IOANNIS GKINALAS
S/N: 13841616

Contents

Cover image: "OV5647 Camera Board /w M12x0.5 mount LS-40180 Fisheye Lens for Raspberry Pi" - www.zigobot.ch. accessed 15 Nov. 2016, < https://dlscorp.com/wp-content/uploads/2018/07/OV5647-PiCAM-M12-RevC-angle-1.jpg>. Distributed under Free Public Domain licence

Abstract

The current paper is the interim report describing the intended process for developing a low-cost portable surveillance system based on a Raspberry Pi. The project is part of the final year project module of the BSc computer Science course at the University of Brighton, for the academic year 2018 – 19.

Aim

Developing an independent surveillance system (CCTV) based on a Raspberry Pi

Objectives

The system should be able to:

- Record images and store the recordings on a remote location/cloud
- A user should be able to view the recordings based on set time-period on a web-based visual dashboard
- The development process is to be recorded in a report, including the methodology, any implications and a student's learning reflection

Potential applications

- Surveillance and security for:
    - External and internal area of properties such as homes and business
    - Remote locations away from buildings and power source, such as: a field or a remote street
    - internally and/or externally of vehicles or other moving objects, such as: Buses, taxis, aeroplanes and flying drones
- Broadcasting images/recordings from special events or scenic locations, excluding live broadcast, such as "live cam"

Stakeholders (clients)

Users

Individuals or business

Project Supervisor

Saeed Malekshahi Gheytassi

Methods of communication

With supervisor

- Regular live meetings (frequency to be agreed) at Mr Malekshahi's office
- Via university's SharePoint Ci301 Surveillance system 2018 site, for files and general information sharing, including emails
- The code and any relevant document will be stored in a repository (the name and link is to be announced by the student) on the version-control web-based platform **GitHub**

Installation Standards & procedures

Hardware

- Raspberry Pi, a single board computer: To be provided by supervisor/university
- Camera module for Raspberry Pi: To be provided by supervisor/university or to be bought
- SD card for storing operating system, operating algorithms and actual recordings in Raspberry Pi: Also to be provided by supervisor/university or to be bought
- Server (SharePoint/Microsoft cloud): Already provided by university
- Computer for accessing hardware, coding and testing: Provided by university in labs and library

Software

- IDE for writing and editing the programming code, such as Notepad++ or Eclipse. These are provided by the university in labs' computers.
- Additional IDEs for depending functionality, such as the front-end dashboard developed in HTML/CSS and java script
- The Raspbian Operating system and relevant algorithms for the Raspberry Pi and its module(s), such as Computer Vision for managing images in Raspberry Pi and mjpeg streamer to stream video over HTTP.
  This software is distributed under the free for use licence based on the GNU General Public License or
- Software, such as Putty, for controlling the Raspberry Pi remotely from a computer via SSH/Telnet

Evaluation

Success Measurement

A successful scenario could be the following:

- While the camera module on a raspberry Pi is turned on, consequently,
- Photographs are recorded and stored in a cloud/server, then
- The user should be able to initiate the dashboard (ie: on a web-page) and be able to request and see the footage from a particular period of time of their choice.

Deadlines

The project with any deliverables should be complete by the set deadline date on 09 May 2019, however deadlines for interim individual tasks will be set in the project management documentation, such as the Gantt Chart

Deliverables

The main basic components are divided in **software** application(s) and **hardware** installation(s)

Software

This should be comprised by three parts: back-end, middle layer and front-end**.** As per **MVC model** as taught in [relevant modules]

Back-end

The code running on Raspberry Pi, with functionality of:

- Recording the footage
- Uploading data to a remote location (database or a cloud storage)

Middle layer / Controller

A controller, linking front-end and back-end, with functionality of:

- Sending requests (data queries) from front-end to back-end, and
- Respectively return the corresponding data back to front-end for display

Front-end

The dashboard where the user may see the footage, with functionality of:

- Selecting a time period
- Displaying the corresponding footage

Extensibility

- Controlling hardware via front-end, such as:
    - Turning it on or off
    - Change the angle of the camera, aiming in different location
    - Selecting input from multiple cameras
- Implementing motion detection algorithm so that the recording happens only during some action is performed at the monitored area, saving energy and disk space for static images

Schedule of activities

Planning & project management

- List of tasks/actions
- Budgeting time for tasks and setting deadlines
- Creating Gantt Chart with tasks and time-frames

Modelling

- Designing / planning components and their relations (class/components diagram)
- Modelling objects/data

Researching

- Examples of similar applications
- Use of hardware (Raspberry Pi and camera component) and its IDE
- Program / code applications for operating the camera
- A single IDE able to host all different programming languages/platforms required for development, such as for:
    - Back-end: Raspberry Pie programming language (C++ or java)
    - Front-end dashboard(HTML, CSS, JavaScript)
    - Database/storage
- Whether footage should be recorded as a daily video or still images and later composed to animation for display
- Whether should store photographs in a cloud storage (as files) or in a database on a server (such as Tomcat)
- Ethical issues the application may have on humans and the environment

Methodology

Mainly Agile with sprints based on the tasks as described in the Gantt Chart including RAD (Rapid Application Development) with prototypes

Risk analysis & potential problems

- Finding an IDE which, as described in research, can support all individual platforms (such as C++ or java, HTML, CSS, JavaScript, and server/database)
- Developing the front-end components, may be develop a steap learning curve, mainly due to student's limited experience and knowledge of programming in JavaScript
- Developing functionality in for:
    - Raspberry Pi functionality in Python, which is the default programming environment
    - Storing the pictures from the Raspberry Pi to a server/cloud
    - Authenticating at the cloud storage for storing and reading the recordings
    - Composing the animation required from pictures taken
    - Linking/connecting the individual parts: front, middle and back

Relevant Modules

Part of the resources are also the modules taught in the course which allowed the student to develop the necessary skills and knowledge. In particular, the modules that have a bearing on the project are:

Module	Developed Skills
• CI101 Programming • CI228 Object oriented software architecture, design and implementation • CI213 Intelligent systems • CI346 Programming languages, concurrency and client server computing	Object oriented programming
• CI284 Data Structures and Algorithms	Designing and implementing complex and sophisticated code
• CI283 Operating Systems	Understanding low level software/hardware interaction and programming in both low and high level languages

• CI116 Embedded Architecture and Programming • CI209 Computer Systems Architecture • CI334 Mobile Computer Engineering	Programming in low level languages
• CI143 Introduction to requirements analysis • CI222 Project planning and control	Requirements and project planning
• CI164 Professional and Study Skills • CI152 Computers in Context	Performing academic research and writing
• CI141 Human Computer Interaction	Designing interactive applications
• CI135 Introduction to Web Development	Front-end software development, including JavaScript programming
• CI107 Mathematics • CI231 Logic and Formal Specification • CI311 Specification and Refinement	Software development planning and design
• CI102 Introduction to databases	Manage and develop databases and servers, including SQL programming
• CI236 Integrated Group Project in Computing • CI282 Placement learning	Software development life cycle and practice of all the above skills

Annotated bibliography

Existing research

Menezes, V, Patchava, V, Gupta, M, (2015). **Surveillance and monitoring system using Raspberry Pi and SimpleCV**. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). Noida, India, 8-10 Oct. 2015. Danvers, MA, USA: Institute of Electrical and Electronics Engineers (IEEE). 1276 - 1278.

A report on how the authors developed a surveillance system using Raspberry Pi and computer vision with SimpleCV for detecting motion without need of light, and streaming a live feed of the recording using MJPG Streamer.

S, Prasad, 2018. **Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor**. (IJCSIT) International Journal of Computer Science and Information Technologies, [Online]. 5 (6), 7107-7109. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.658.6805&rep=rep1&type=pdf [Accessed 15 November 2018].

Prasad designed and developed a surveillance monitoring system
With Raspberry pi and Passive Infrared (PIR) sensor for motion detection while also streams the footage to a smart phone with a 3G Dongle and displays it using web application.

Nguyen, H, Loan, T, Mao, B, Huh, E, (2015). **Low cost real-time system monitoring using Raspberry Pi**. in: 2015 Seventh International Conference on Ubiquitous

and Future Networks. Sapporo, japan, 10 August 2015. Danvers, MA, USA: Institute of Electrical and Electronics Engineers (IEEE). 857 - 859

Report on developing a low cost real-time system monitoring with motion detection and transmitting the footage over network, using a Raspberry Pi.

Abaya, W, Basa, J, Sy, M, Abad, A, Dadios, E, (2014). **Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV**. in: 2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). Palawan, Philippines, 12-16 Nov. 2014. Danvers, MA, USA: Institute of Electrical and Electronics Engineers (IEEE). 1 - 6

A report on developing a security camera with night vision capability using Raspberry Pi with RPI sensor and Open Source Computer Vision (OpenCV) with motion and smoke detection capabilities.

Sub-areas of the topic

Monk, S. (2016). **'1.14 Installing the Rasbery Pi Camera Module' in *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions***. 2nd ed. Sebastopol, California: O'Reilly Media. Page 22

Guided instructions on how to install the camera module on the board.

Monk, S. (2016). **'8.3 Using a Raspberry Pi camera Module for Computer Vision Problem' in *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions***. 2nd ed. Sebastopol, California: O'Reilly Media. Page 182

Guided instructions on programming and using the raspberry Pi camera Module including the Computer Vision (OpenCV) software.

Monk, S. (2016). **'8.6 Motion Detection' in *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions***. 2nd ed. Sebastopol, California: O'Reilly Media. Page 189 – 193

Guided instructions on capturing images only when detecting motion with Raspberry Pi and its camera module using the software OpenCV

GNU Project - Free Software Foundation. 2018. GNU General Public License, version 2.0. [ONLINE] Available at: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html. [Accessed 15 November 2018].

Complete official description of the GNU General Public License by the Free Software Foundation, for licencing open use of software such as for Raspberry Pi's operating system, Rasbian.

rasbian.org - Raspberry Pi Foundation. 2018. Rasbian. [ONLINE] Available at: https://www.raspbian.org/. [Accessed 15 November 2018].

Rasbian's operating system licence statement by Raspberry Pi Foundation

Relevant Research

Practical Tutorials

Pi My Life Up. (2015). Build a Raspberry Pi Security Camera Network. [Online Video]. 20 June 2015. Available from: https://www.youtube.com/watch?v=P-cg2WMaPZ0. [Accessed: 15 November 2018].

Video tutorial on building and setting up a Raspberry Pi Security Camera Network

Hacker Shack. (2015). How to Make a Smart Security Camera with a Raspberry Pi Zero. [Online Video]. 19 September 2018. Available from: https://www.youtube.com/watch?v=Y2QFu-tTvTI. [Accessed: 15 November 2018].

Video tutorial on building a Security Camera with a Raspberry Pi Zero Including creating a housing case box for the hardware using MDF board

Software sources

Acme Systems srl. 2018. WebCam: Using mjpeg streamer to stream video over HTTP. [ONLINE] Available at: https://www.acmesystems.it/video_streaming. [Accessed 15 November 2018].

Instructions on using **mjpeg streamer** for streaming video over HTTP

Sight Machine, Inc. 2018. Computer Vision platform using Python. [ONLINE] Available at: http://simplecv.org/. [Accessed 15 November 2018].

Instructions on using **SimpleCV** for implementation computer vision with Raspberry Pi