Specification of Requirements

Project Kitchen Occupation

TSBB11 HT 2013

Version 1.1

Status

Reviewed	–	2013–11–27
Approved		

2013–11–27

Project Kitchen Occupation

Bilder och Grafik CDIO, HT 2013
Department of Electrical Engineering (ISY), Linköping University

Participants

Name	Tag	Responsibilities	Phone	E-mail
Mattias Tiger	MT	Project manager	073–695 71 53	matti166@student.liu.se
Erik Fall	EF	–	076–186 98 84	erifa226@student.liu.se
Gustav Häger	GH	System integration	070–649 03 97	gusha124@student.liu.se
Malin Rudin	MR	–	073–800 35 77	malru103@student.liu.se
Alexander Sjöholm	AS	–	076–225 11 74	alesj050@student.liu.se
Martin Svensson	MS	Documentation	070–289 01 49	marsv106@student.liu.se
Nikolaus West	NW	Testing	073–698 92 60	nikwe491@student.liu.se

Homepage: TBA

Customer: Joakim Nejdeby, Linköping University, Origo 3154
Customer contact: 013–28 17 57, joakim.nejdeby@liu.se
Project supervisor: Fahad Khan, Linköping University, fahad.khan@liu.se
Examiner: Michael Felsberg, michael.felsberg@liu.se

Contents

List of Figures

Document history

Version	Date	Changes	Sign
0.1	2013–09–09	Initial draft	MS
0.2	2013–09–18	Scrum adaptation	MT
1.0	2013–09–24	Final Document	All
1.1	2013–11–27	Revision of several requirements due to changed conditions combined with the realization that depth information is needed to reach good enough performance.	MS, EF, MT

1 Introduction

There exists many places in society where the degree of human occupancy and movement flow is desirable to know as basis for decision making. Such data answers if it is necessary to build more rooms and provides knowledge of actual user or consumer patterns. Example usages are measuring reasource usage of public spaces, or which part of a store that attracts most people. It provides vast opportunities in resource management, marketing, sales and scheduling. There exist some plausible solutions to estimating the number of people at a location such as using cell phones or motion detectors, but this project aims at an image based approach with the possible benefits of being both cheaper and more robust.

1.1 Background

Today Linköping University has many places with similar functionality, e.g. student kitchens where students are provided with the ability to warm food brought with them. Linköping University has several such kitchens all over its campuses. Critics claim that there are too few student kitchens with microwave ovens and that the existing ones usually are overcrowded. That all kitchens are overcrowded at the same time has not been confirmed by sample inspections. One standing hypothesis is that students don't know where all the kitchens are nor that they want to risk going to a kitchen in another building in case that is full as well.

Linköping University has an ongoing project with the purpose of enabaling the students to see the usage of some of the schools resources (e.g. group rooms) online. The aim of this project is to supply that system with data regarding the usage of student kitchens. It will provide all students with the ability of visualising the crowdedness of each kitchen, thus providing them with the means of finding the closest, least occupied kitchen available.

1.2 Involved Parties

Three parties are involved:

- LiU IT, the Division for IT servidces at Linköping University.
- Computer Vision Laboratory, Department of Electrical Engineering, Linköping University.
- A group av students taking the course TSBB11 2013, listed in the *Participants* table, page (ii).

1.2.1 Customer

Liu IT, represented by Joakim Nejdeby, CIO at Linköping University.

1.2.2 Supervisor

Ph.D Fahad Khan at the Computer Vision Laboratory, Department of Electrical Engineering, Linköping University.

1.2.3 End users

Students at Linköping University that want to use the student kitchens.

1.3 About this document

This document contain the requirements of the project. It is divided into different modules or aspects, each with further subdivisions, all containing explanatory text and functional requirements. Each functional requirement is placed in a table of the form showed below.

Req.	Description	Type

1.3.1 Requirement priorities

Each requirement has three different types, the meaning of each one is presented below:

1. type one constitutes a mandatory requirement, meaning this feature has to be fulfilled by at the time specified ion the description. If no time is specified, the requirement has to be fulfilled by the time of the final delivery (see section 8).
2. A requirement with type two is a requirement to be met if extra time is available.
3. A type three requirement is more of a suggestion on how to improve the system even further after the final delivery.

2 System Overview

The system consists of two parts, one that is designed to run in real-time on a laptop or other mid-end processing device, complete with GUI for debugging and parameter tuning. An other, more slim version designed to run, in real-time on a Raspberry Pi[1] computer, with a Raspberry-Pi camera or a Microsoft Kinect is also delivered. In the former case all processing of data takes place on the processing unit and is presented to the user via the debug-GUI. The Raspberry Pi-system, image processing is performed locally on the board with information about people entering/exiting being transmitted via Ethernet to receiving device. Below is an overview of the system.

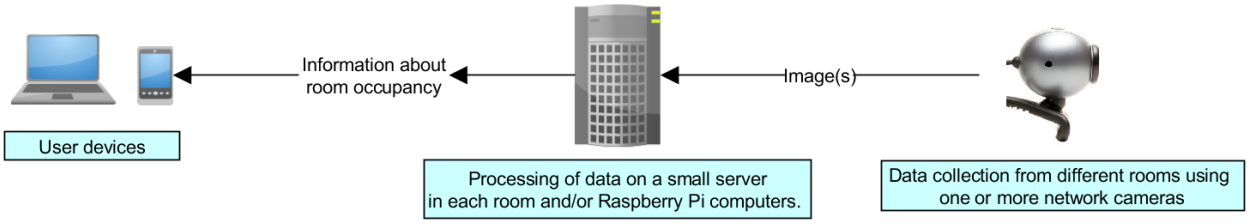

Figure 2.1: *A simplified overview of the system*

2.1 Rough description of the systems

The camera(s) used to collect data are connected to the processing unit via Ethernet, or USB in case where one processing unit per room is used. For integrity reasons, the network connecting the cameras to the local processing unit needs to be private and through cable. For the Raspberry Pi system, a special Raspberry-Pi camera board, or a Microsoft Kinect is used. The main program collects data from the cameras in a room to perform an estimation of room usage intensity, which is then presented to the user in an understandable format, e.g. estimated waiting time or current amount of people in the room.

[1]The words "Raspberry Pi" and the Raspberry Pi logo are trademarks of the Raspberry Pi Foundation

2.2 Components

The main components of the system are the cameras, the image processing unit(s), and the result presentation page.

2.2.1 Hardware

The system runs on both Raspberry Pis and a regular laptop. The laptop system runs on any OpenCV-compatible camera as well as the Microsoft Kinect, whereas the Raspberry Pi system requires the Raspberry Pi camera board or a Microsoft Kinect unit as an input. For a more detailed description of the hardware requirements, see section 3.

2.2.2 Software

In the case of regular cameras being used, the software will be running on a server where both image processing, estimation of queue size and waiting time takes place. If the Raspberry Pi:s are used, image processing takes place locally on each board. For a more thorough description of the image processing and estimation programs, see section 4.

2.3 Usability and installation

In order to create a system that is cheap to use and install, it needs to be easy to set up, which is why a user's manual and an installation/calibration program is provided with the system if necessary. As for the usability, relevant data is presented to the user on a web page or via the Debug-GUI in case of further development (see section 6).

3 Hardware

Since the system is to be able to run on two different platforms, the hardware requirements differ a lot depending on which system being considered. In the case of Raspberry Pi computers being used, all cameras each have their own Raspberry Pi processing unit where image processing takes place. In case of a central image processing unit (laptop, server or desktop computer), image processing from all cameras in the room take place on this central unit.

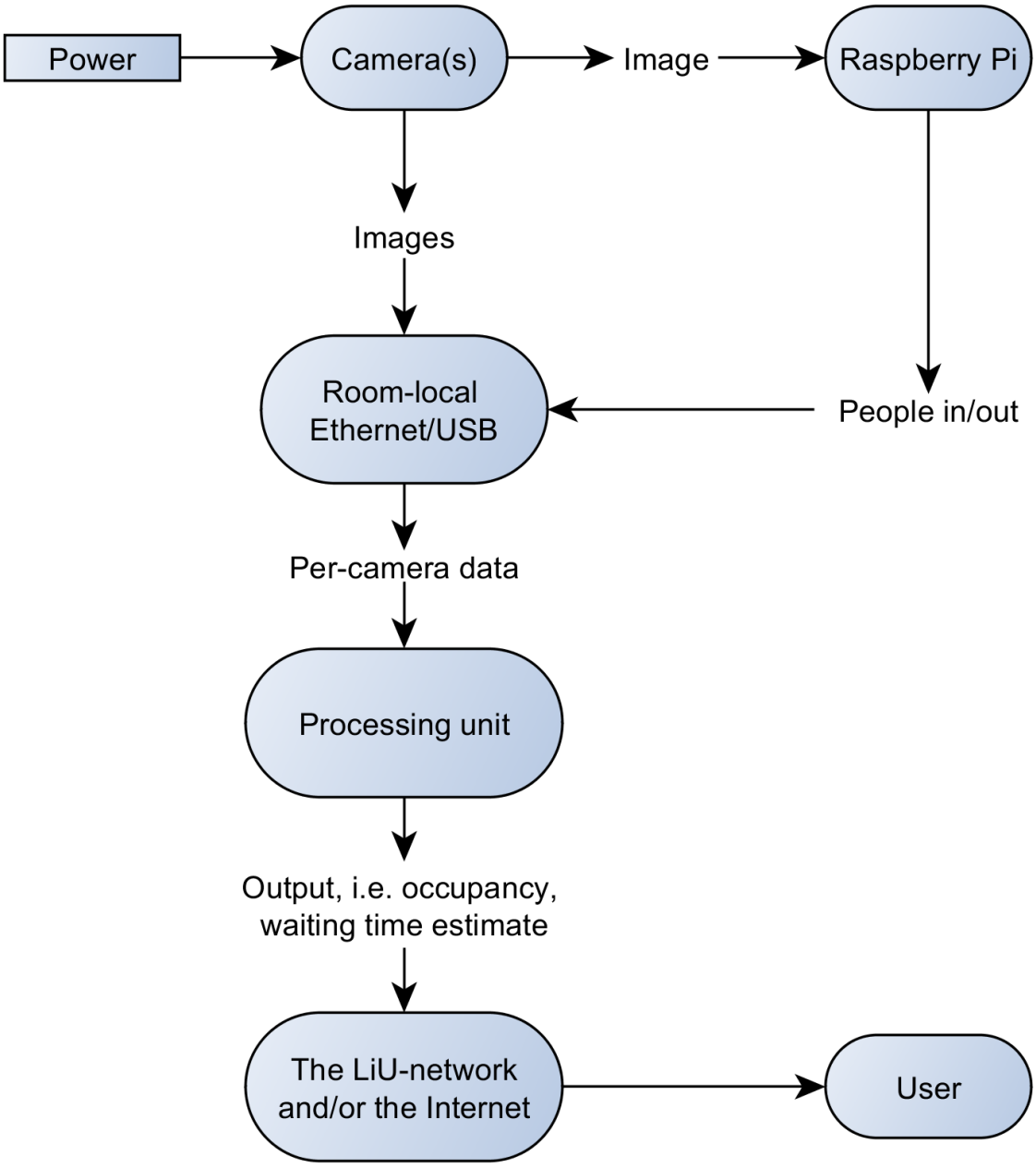

Figure 3.1: *Flowchart displaying data flow between hardware modules*

3.1 Limitations

The hardware is limited by budget, Internet connection and the source of power. The cost is limited to approximately 15.000 SEK per room, including installation costs. The budget limits performance of the cameras e.g. resolution and number of cameras. The connection from the cameras to the processing unit(s) needs to be stable and have a bandwidth good enough for sending live video from the cameras. Additionally, no image data may be transmitted over public networks.

3.2 Hardware Requirements

Req.	Description	Type
3.1	The system uses network cameras powered via Ethernet. **Revised 2013-11-27:** The system uses depth sensors connected to a laptop (or similar mid-end processing device) via network or USB.	**1**
3.2	The system can operate using high resolution (>2 Mpixel) cameras **Revised 2013-11-27:** The can operate using depth sensors, in this case the Microsoft Kinect.	**1**
3.3	Lower resolution cameras can be used **Revised 2013-11-27:** Stereo cameras can be used.	**2**
3.4	The application can run using the processing power provided by the costumer. **Canceled 2013-11-27:** No processing power will be provided by the customer within the delivery time.	–
3.5	The application can run using a mid-end processing device (e.g. a laptop or desktop computer). **Revised priority 2013-11-27:** The cancellation of previous req.(**3.4**) causes priority to change **from 2 to 1.**	**1**
3.6	The application can run using a low-end processing device. **Revised priority 2013-11-27:** Revised due to a request from customer that the system is able to run on a Raspberry Pi. Priority changed **from 3 to 2.**	**2**

4 Software

The software part of the system performs all the image processing and analysis needed to detect usage of a room. It will also be cablable of predicting future room occupancy degree based on historical usage.

A website for the project will be published using the wordpress CMS or using GitHubs webpage services, it will describe the project and its participants.

4.1 External dependencies

The software is written using the OpenCV library to handle the image processing, and when possible interfacing with cameras. Any visualization and debugging tools are written using the Qt framework. OpenNI is used for accessing the Kinect camera. An HTTP library is used to handle communication with LiUs REST API. Cmake and any C++11 capable compiler can be used to compile the source code to binary.

The website will be hosted on a standard LAMP stack.

4.2 Compatibility

The software is possible to build for most major platforms (Windows/OS X). A REST API is used to communicate the results.

4.3 Limitations

For integrity reasons no personally identifiable information can be stored or reported by the system. The processing is done locally for each camera on a Raspberry PI or on a remote computer connected via network or USB. Depending on which of these hardware configurations is used, the computational power will differ and therefore limiting or enabling the use of computational-heavy algorithms.

4.4 Software Requirements

Req.	Description	Type
4.1	The system runs on Windows based plattforms	**1**
4.2	The system runs on OS X	**1**
4.3	**Added 2013-11-27:** The system runs on Raspberry PI	**2**
4.4	The system is modular with respect to the camera manufacturer and/or network API. **Revised 2013-11-27:** The system is modular with respect to any depth sensor and network API	**1**
4.5	The system must not store image data	**1**

5 Performance

The performance of the system refers to what problems the system solves and the degree to which it solves them. The problem that the system solves is giving indications to how long the wait will be for use of microwaves in the student kitchen. The simplest indicator for this is simply the amount of people in the kitchens. This is made slightly more complicated by the potential presence of more than one door into the room. The amount of people in the room is also easily verified by hand, meaning that performance with regard to this indicator is easily defined and quantified. Other indicators of the length of waiting are detection of a queue, rough classification of the severity of the queue, and a directly estimated waiting time. The system performance on these indicators are all increasingly both hard to define and measure. The inclusion of methods to evaluate and test these more difficult indicators are therefore necessary.

5.1 Reliability

The system should ideally be able to perform reliably under varying lighting conditions, while also being able to handle people with varying hair color and whearing a large varaiaty of different clothes, jackets, hats. It is preferable that the system also handles bags, trolleys etc. without counting them as extra people.

5.2 Quality control

The system is thoroughly tested throughout development to ensure high quality. The core computer vision functionality is also continuously tested against a test data set that is never used to train or tune any algorithms.

5.3 Performance requirements

The performance requirements listed below assume that it is possible to have cameras placed over each door. Removing this assumption results in increasing the type number by one (e.g. from 1 to 2).

Req.	Description	Type
5.1	The system is able to count the number of people entering and leaving the room, where the room has one door.	**1**
5.2	The system is able to count the number of people entering and leaving the room, where the room has two doors.	**1**
5.3	The system keeps track of the amount of people in the room at any one time, given that it can count the number of people passing through each door.	**1**
5.4	The system knows if there is a queue to enter the room. **Canceled 2013-11-27:** Redundant because of next requirement in the table.	**1**
5.5	A rough classification of the queue size/severity is presented by the system, where the room has one door.	**1**
5.6	A rough classification of the queue size/severity is presented by the system, where the room has two or more doors.	**2**
5.7	The system gives a model based estimate of the waiting time that is more informative than the rough queue classifications.	**2**
5.8	The system can handle daily variations in lighting conditions such that other performance metrics are not affected. **Clarified 2013-11-27:** The system can handle daily indoor lightning conditions such that other performance metrics are not widely affected.	**1**
5.9	The system handles sudden changes in lighting (e.g. a blackout) without crashing and keeping track of potentially induced knowledge gaps.	**2**
5.10	The system is tested against a data set that covers a wide variety of the most common cases. **Revised 2013-11-27:** The system is tested against a data set that covers the most common cases.	**1**

6 Usage and Installation

This section describes what is required by the system in terms of usage and installation. It also covers what is wanted by the user.

6.1 Installation

The installation consists of the placement of one or more cameras, connection of these via Ethernet to a computer, and installation of the software on that computer. The cameras should be placed as described in the user manual with focus on vision rather than precise positioning. The connection via Ethernet is completely provided by the user.

Once installed the system needs a calibration which is preformed by the user in the calibration program.

The installation and calibration process is described in detail to the user in the user manual.

6.2 Usage

When the system is properly installed it can be maneuvered by the user through the interface. The user can now start adding cameras together from the interface. Now the system will start keeping track of the room and continuously provide the user with occupancy statistics about the room.

6.3 Maintenance

The system should be able to run continuously once started, without further maintenance. Additional cameras can not necessarily be added on the fly.

6.4 Continued development

The user interface is initially not prioritized as this is a project mainly in computer vision. This will leave some GUI features open for future development.

6.5 Operational requirements

The user is assumed to have some minor computer skills, but no knowledge whatsoever about computer vision.

Req.	Description	Type
6.1	The installation proces requires a short manual and no knowledge about advanced computer vision	**1**
6.2	Calibration of the system is performed via a calibration program	**1**
6.3	The system is self-calibrating	**2**
6.4	Software for adding new cameras and/or rooms is provided with the system. **Revised 2013-11-27:** Only single-room - many sensors support is provided.	**1**
6.5	Results are presented on the project group webpage	**1**
6.6	System is avaliable as an App on AppStore/Android Market	**3**

7 Documentation

The following documents are produced along the course of the project.

7.1 Project plan

The outline of the project is described in the project plan. It specifies the nature of the project, responsibilities, resources available, the organization of the project and development methods. It also contains the preliminary Product backlog with preliminary priorities and estimated finish dates based on the priorities.

7.2 User's manual

To help the user to install and use the product, a user's manual is presented. It is a description of the installation and usage of the product.

7.3 Scrum review document

After each sprint a Scrum review document is produced containing the sprint backlog together with the result of the sprint and a review.

7.4 Technical report

At the end of the project the result of the project is documented in a technical report. This report is a detail description of the different aspects of the project, such as hardware and software solutions, limitations of the system and further developments.

7.5 Documentation Requirements

Req.	Description	Type
7.1	A project plan providing an outline of responsibilities and development methods has to be presented to the supervisor	**1**
7.2	At the end of the project a technical report is delivered to the customer and course examiner	**1**
7.3	A user's manual will be delivered with the technical report	**1**

8 Delivery

A partial system will be delivered at the Mid-term checkpoint and the full system along with documentation is delivered at the final delivery deadline.

8.1 Mid-term checkpoint

A partially working system is delivered (What this means will be clarified here once the project plan is finalized).

8.2 Final delivery

A kitchen occupancy software system, fulfilling at least all requirements of type 1, is delivered to the customer. Full documentation specified in section 7 is delivered to the customer. A project web page is on-line with a short introduction to the project and a video demonstration of the system.

8.3 Delivery dates

This document	**2013-09-24**
Project plan	**2013-09-24**
Sprint review document	**At the end of each sprint**
Final product	**2013-12-13**
Technical report	**2013-12-13**
User's manual	**2013-12-13**
Final presentation	**2013-12-19**