Sprint reviews

Project Kitchen Occupation

TSBB11 HT 2013

Version 1.0

Status

Reviewed	–	2013–12–13
Approved		

2013–12–13

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	2013–12–08	Initial draft	MS

1 About this document

This document, together with the attached time report spreadsheet and the Github commit history (`https://github.com/GroupDenseKitchen/KitchenOccupation/graphs/contributors`), constitutes the project log and is a description of how work has progressed during the project. Within this document are plans and reviews of all project sprints, complete with tables and comments to the results of each sprint.

1.1 Changes from the project plan

Compared to the project plan, the work methodology used is for the most part exactly according to the plan, with the exception of test-driven development, as writing unit-tests was considered to be too time-consuming since a large majority of the group had no experience of this development method. Functionality and quality was instead ensured by thorough testing use of Github pull requests with code reviews. As far as the time plan is concerned the project direction changed drastically in the middle of sprint 3 when the mono-camera approach was abandoned, however due to the agile SCRUM development method and a nice software framework, most of the initial intended functionality was in place at the time of delivery.

1.2 About the time spreadsheet

Attached with this document is a time spreadsheet with a complete presentation of how the work has progressed and how and what each group member has contributed. The notation **SX:Y(Z)** has the following meaning:

- **X** is the number of the sprint that the item belongs to.
- **Y** is the item number within that sprint
- **Z** is the time spent working on the item the two p other numbers specify.

2 Review of Sprint 0

Period: 2013-09-17 – 2013-09-30
Est. time: No estimate was made

2.1 Sprint backlog

This sprint is the first one for the project and is thought of as a way of testing the process, which is why no time estimations were made. The main purpose of this sprint was to create frameworks and routines for the continued development.

2.2 Sprint plan table

Item	Description	People
1	Requirement specification: Fix the remaining issues with the requirement specification.	EF
2	Project plan: Add the product backlog, due dates and priorities.	GH, MR
3	Project plan: Document and process explanation.	MT, NW
4	Documentation: Create a template for this document	MS
5	Data collection: Gather a first batch of test data using USB cameras.	MR, AS
6	Data collection: Find a suitable way of labeling the different video sequences	EF, MR, AS
7	Code (general): Create code skeleton, mainloop and interfaces between modules.	MT, NW
8	Code (general):Set up a cmake build system for the entire project.	GH, MS
9	Code (general): Create a code standard.	MS, MT
10	Documentation: Set up DOxygen	GH, MS
11	Testing: Set up an outline for some form of automatic testing.	GH, NW
12	Technical documentation: Create a template for the final technical documentation.	AS, MS
13	Sprint Review Document: Create a template for the Sprint review documents.	MS

2.3 Result

The result of this sprint can be considered good since all items were completely finished on time. What needs to be improved by in the next sprint is to use the item numbers for the planning meetings on the SCRUM board and when naming issues on git. The reporting of time also has to be improved in order to be able to distribute workload evenly among group members.

2.4 Sprint result table

Item	Result	People
1	DONE, but no feedback was recieved from supervisor.	EF
2	DONE, but no feedback was recieved from supervisor.	GH, MR
3	DONE, but no feedback was recieved from supervisor.	MT, NW
4	DONE, resulted in a new backlog item (1).	MS
5	DONE, but FoV was too narrow using the USB camera.	MR, AS
6	DONE, CAVIAR GUI is used for labeling.	EF, MR, AS
7	DONE, resulted in two new backlog items (2 and 3).	MT, NW
8	DONE.	GH
9	DONE, see the project wiki page at github.	MT, MS
10	DONE, see the project wiki page at github.	GH, MS
11	DONE, however a full system test is not complete as this item was only regarding the framework.	GH, NW
12	DONE.	AS, MS
13	DONE.	MS

2.5 New Backlog Items

The first sprint resulted in three backlog items, all of which are listed and described below

2.5.1 Review collection

Adding a template for the review collection, which is a collection of these documents, one for every sprint that are to be submitted at the end of the course.

2.5.2 Algorithm interface

The algorithm interface in the image processing and the content of the FrameList class.

2.5.3 DOxygen comments

DOxygen comments need to be added to the code skeleton.

3 Review of Sprint 2

Period: 2013-09-30 – 2013-11-04
Est. time: No estimate was made
Time spent: 227 hours.

3.1 Sprint backlog

The main focus of this sprint is to get the simplest possible system up and running on pre-recorded test data. Unfortunately no time was available on the planning meeting to perform time estimations, which is why no time estimates are available for the different sprint items. The ending date of this sprint was postponed 2 weeks due to exam periods, since not enough work would be made during these two weeks in order to justify an entire sprint.

3.2 Backlog items from previous sprints

These are the items created in the previous sprint that were carried over directly to this one

3.2.1 Algorithm interface

This item was created during sprint 0, and is represented in this sprint as a part of item number **3**.

3.2.2 DOxygen comments

This item was added to the product backlog in sprint 0 and is here represented by sprint item number **4**.

3.3 Sprint plan table

Item	Description	People
1	Collect more test data, several data sets with varying difficulties using a single camera above a rooom entrance are desired.	AS, EF
2	Label the new data (using the CAVIAR labeling program).)	AS
3	Add more data structures to the code skeleton, including an algorithm interface.	GH, MT
4	Add DOxygen comments to the code skeleton.	MT
5	Load video file(s) into network module, making it possible to retrieve individual frames.	MS
6	Implement background model and segment background/foreground (Raw image in and foreground binary image mask for the pixels that are "moving" out. (Use OpenCV background subtractor class)	EF, NW
7	Foreground labeling. Find distinct regions in a binary image mask and describe them with position and AABB. Add regions as "objects" to current Frame.	EF, MS
8	Track objects in the frames using the information received from the foreground labeling and estimate the net flow of objects.	EF, MT
9	Debug program: Display the output of different modules as well as intermediate steps in the image processing module for each Frame.	AS, NW, MR
10	Debug program: Step between frames and intermediate algorithm steps.	AS, NW, MR
11	Debug program: Read a simple config file (select data set).	AS, NW, MR
12	Debugging: Wrap debug code and make it activateable	GH, MT
13	Debugging: Read and use labeled test data as ground truth for performance evaluation	MR, MS
14	Create a config file system	MS

3.4 Result

The result from this sprint is good, as all of the items except for one was finished, providing a great framework for more advances algorithms to be added. The most notable thing not finished is the net flow estimation and the fact that the system still does not run in real-time on a camera. The total amount of hours spent on specific sprint items adds up to 126 hours with meetings excluded. With meetings and technical support hours the total number of hours amount to around 200 hours.

3.4.1 Solutions we ended up using

Currently a mixture-of-Gaussian background model is used together with morphological erode/dilate to identify foreground regions. These regions are then tracked using a local greedy tracking algorithm. The tracking algorithm features candidate object that have to fit certain criteria (currently lifetime) in order to be considered real objects.

3.4.2 What did we do well?

Most of the project work was done well, especially the short meetings, which were short and effective, resulting in new problems being solved efficiently. The most important achievement completed during this sprint is the debug and algorithm platforms. The debugging program allows easy visibility and profiling of each intermediate algorithm step, together with an easy interface for implementing new algorithms in a rapid fashion. Group members have also developed their skills and are able to use the tools (mainly Github) much more efficiently than in the previous sprint, hopefully reducing the technical support hours spent by group members.

3.4.3 What improvements are necessary?

The most important improvement needed in the next sprint is how group members communicate when making design decision. Some unnecessary parallel work was made due to bad communication. Persons in the group with less hours worked need to spend more time working on the project. Because of this the time reporting has to improve significantly in the future.

3.5 Sprint result table

Item	Result	People
1	Done, multiple data sets with different difficulties were collected.	AS, EF
2	Done, labeled with ground truth data for tracker performance evaluation.	AS
3	Done, code skeleton is completely finished.	GH, MT
4	Done, except for Object.hpp, because of dependency issues. Backlog item was created.	MT
5	Done, still only tested for reading from one single file.	MS
6	Done, however, the algorithm is much too slow and needs improvements.	EF, MT
7	Done.	EF, MT, MS
8	Done, with the exception of net flow estimation.	EF, MT
9	Done.	AS, NW
10	Done. (forward only)	AS, NW
11	Done.	AS, NW
12	Dismissed, deemed unnecessary at the moment	–
13	Done.	MR, MS
14	Done.	MS, MT

3.6 New Backlog Items

This sprint resulted in a few new backlog items being created during the sprint, these items are listed below.

3.6.1 Select video files and/or cameras from Debug GUI.

In order to make testing easier and faster, by not having to restart the program to load a new file and/or view a new camera.

3.6.2 Object class

Make sure the Object class meets the code standard.

3.6.3 Improve foreground segmentation

The current foreground segmentation is both much too slow and generates a lot of false positives (pixels falsely classified as foreground). This both of these issues have to be remedied in order to achieve desired performance.

3.6.4 Ground truth labeling

Currently there is no way to evaluate system performance with respect to how well it performs when counting people entering or exiting the room. Program code to both perform this labeling of this data as well as evaluation code has to be added to the system.

4 Review of Sprint 2

Period: 2013-11-04 – 2013-11-18
Est. time: 295 hours. (Roughly 42 hours per person, meetings (6 hours per person) not included.) Time spent: 245 hours. (Meetings included.)

4.1 Sprint backlog

The main focus of this sprint is to get a minimal case implementation that meet all requirements in the specification of requirements. The initials of the person responsible for each specific sprint item is shown in boldface in the rightmost column of the table.

4.2 Backlog items from previous sprints

These are the items created in the previous sprint that were carried over directly to this one

4.2.1 Improve foreground segmentation

Both performance and computation time of the foreground segmentation needs to improve in order to make it useful. This item was created during **sprint 1** and is represented in this sprint as item number **10**.

4.2.2 Ground truth labeling

In order to be able to present projects result with some form of performance measurement ground truth data, as well as evaluation code has to be created. This item was created during **sprint 1** and is represented by sprint item number **4**.

4.3 Sprint plan table

Item	Description	People
1	Explain and visualize the system design and how to continue building on it. The explanation is to be published on the group Github pages.	**NW**, MT
2	Installation/Configuration GUI. Adds possibility to set what image regions should be excluded from the image processing, where do people leave/enter the image plane.	**AS**, MT
3	Gather test data from realistic environments (e.g. CYD-poolen) and label these.	**EF**, AS
4	Add net flow estimation to the statistics module.	**EF**, MR
5	Make the system more robust against illumination. OpenCV settings? Better background model. Shut down dynamic exposure/gain in camera? normalized pixels? (make the system entirely intensity independent.)	**MT**, MS
6	Handle people standing still. (Worst case: rewrite background model. Best case: feed OpenCV with a bit mask.)	**EF**, AS
7	Add queue detection (Requires **6**)	**NW**, GH
8	Gather more information about methods from different computer vision papers.	All
9	Simple classifier of humans from above.	**NW**, GH
10	Improve foreground segmentation, allowing more precise people detection, especially when people are close together.	**GH**, MR
11	More sophisticated error measurement for how probable it is that a previous object and a current one are the same.	**NW**, MT
12	Use a video stream that is in real time from a camera as input to the system	**MS**, AS
13	Update and "renegotiate" requirement specification (Power over ethernet camera no longer an option.)	**MR**, MS
14	Handle one camera over each door to the same room and keep a consistent count of hte number of people in the room (Net flow estimation upgrade, making **4** work using several cameras.)	**EF**, MR
15	Handle multiple video files as separate cameras.	**MS**
16	Collect 3D data set. One set viewing a room with people moving and another with one camera above the door and an other looking towards the door.	**EF**, MR
17	Create system for labeling and reading ground truth of people count. (in/out)	**MR**, MS
18	Finalize the sprint review of sprint 1 and plan of sprint 2 (this docuement and submit these to the supervisor.)	**MS**

4.4 Result

The results from this sprint were not the ones we hoped for as five items are marked as not done, and several others are marked as not tested or have some small issues left. In total 7 out of 16 items are not finalized, which is not good enough. Time estimates were probably accurate, however the amount of time available to group members was exaggerated, hence the difference between the plan and the time actually spent.

4.4.1 Solutions we ended up using

We tried the OpenCV hough circles function with tuning, however it was not good enough for our application, meaning that we will need to implement circle detection on our own. This was also the reason item number 9 was not finished on time. The OpenCV hough circles functon was tried out on images with removed background (using the existing BackgroundModelMOG2 background subtractor.) Point descriptors for simple human classification were also evaluated without success.

4.4.2 What did we do well?

Several relevant papers solving the problem were found. This allowed for good re-organization of the work to both solve the new problems as well as meeting changed demands in terms of available hardware.

4.4.3 What improvements are necessary?

The most important improvement that needs to be made is to get better at working in parallel to try different approaches to the problem, together with prioritizing the tasks in a better way. The most important thing at the moment is to focus on the goal of the project, which is making sure it is possible to deliver a system that can count people passing under a doorway with reasonable accuracy at the time of the deadline.

Item	Result	People
1	Done, but an overview image is missing.	NW, MT
2	Done.	AS
3	Done, but better data need to be gathered as the current data set is out of focus at head level.)	AS, MT, MS, NW
4	Done. (in/out flow is estimated.)	EF
5	Done. Exposure time is now constant and 50Hz noise is removed.	AS, EF
6	Done, with a naive implementation that should be improved once the segmentation improves.	EF
7	Developed, however it is untuned, untested and not inserted into the system, as better tracking/segmentation is required to test this properly.	NW, GH
8	Done. Papers seem very useful, and have been studied.	MS, NW, MR
9	Not done. better data set required.	NW
10	Not done, the OpenCV function does not perform good enough.	GH, MR
11	Slightly better but not good enough to work alone, and will probably have to be combined with some form of human detector.	NW
12	Done.	MS
13	Not finalized, more discussions	MS
14	Not done.	EF
15	Done.	MS
16	Not done.	None
17	Not done. Labeling program is complete, but integration of the evaluation is not complete.	MR, MS
18	Done.	MS

4.5 New Backlog Items

Several new backlog items were created as a result of this sprint, most of them consist of unfinished items that are carried over directly to the next sprint.

4.5.1 Program architecture image

Item number 1 in this sprint. Added to the backlog and is required to provide a good overview of the system to allow for easy continued development after the project is finished.

4.5.2 Queue detection

Needs to be tested and integrated into the project as soon as the requirements to do so are met (stable tracker/segmentation).

4.5.3 Human detector

A high priority item for the next sprint.

4.5.4 Improved foreground segmentation

This item will probably be a direct result of the foreground segmentation.

4.5.5 Better tracker

The tracker needs to be tested and tuned once the people detection performs adequately.

4.5.6 More specific information about project goal and requirements

Also a high priority item. Specification of requirements need to be revised and updated.

4.5.7 Assume that all cameras belong to the same room

This needs to be added to the specification of requirements.

4.5.8 Collect 3D data set

Collect a 3D data set using a stereo camera rig or a Microsoft Kinect.

5 Review of Sprint 3

Period: 2013-11-18 – 2013-12-02
Est. time: 280 hours, meetings included.
Time spent: 311 hours, meetings included.

5.1 Sprint backlog

The main focus of this sprint is to get a minimal case implementation that meet all requirements in the specification of requirements. The initials of the person responsible for each specific sprint item is shown in boldface in the rightmost column of the table.

5.2 Backlog items from previous sprints

These are the items created in the previous sprint that were carried over directly to this one. This week, all of the items below are from the previous sprint (**2**).

5.2.1 Program architecture image

Item number 1 in this sprint. Added to the backlog and is required to provide a good overview of the system to allow for easy continued development after the project is finished.

5.2.2 Queue detection

Needs to be tested and integrated into the project as soon as the requirements to do so are met (stable tracker/segmentation).

5.2.3 Human detector

A high priority item for the next sprint.

5.2.4 Improved foreground segmentation

This item will probably be a direct result of the foreground segmentation.

5.2.5 Better tracker

The tracker needs to be tested and tuned once the people detection has reached adequate performance.

5.2.6 More specific information about project goal and requirements

Also a high priority item. Specification of requirements need to be revised and updated.

5.2.7 Assume that all cameras belong to the same room

This needs to be added to the specification of requirements.

5.2.8 Collect 3D data set

A Microsoft Kinect is now available, and will be used to collect a 3D data set (item number **4**).

5.3 Sprint plan table

Item	Description	People
1	Implementation of circular Hough-based classifier.	**NW**, MS
2	Tuning of circular Hough-based classifier.	**NW**, MS
3	Tracking using optical flow.	**EF**, GH
4	Simple segmentation using optical flow.	**EF**, GH
5	Segmentation using optical flow and super-pixels.	**EF**, GH
6	Remove debugging from raw-image into a debug image	**GH**
7	Collect a top-down view data set with depth using the Microsoft Kinect.	**AS**, MT, MR
8	Calibrate a stereo camera. **Cancelled because of the Kinect-sensor.**	**AS**, MT
9	Use height information from stereo cameras to segment and classify humans.	**AS**, MT, MR
10	Collect a new data set using a camera with focus at head level rather than ground level.	**MS**, NW
11	Integrate evaluation of in/out people count from ground truth.	**MR**
12	Collect an RGBD data set using the Kinect sensor mounted above a room entrance.	**AS**
13	Tuning and merging of integrated parts in the system as well as tuning the entire pipeline.	**All**
14	Handle one camera over each door to hte same room and keep a consistent count of the number of people in the room. (Same as item number 4 in previous sprint, but this time with several cameras.)	**EF**, MR
15	Finalize this document and the review of hte previous sprint.	**MS**
16	Do a complete overhaul of the entire specification of requirements to fit the new, changed, circumstances.	**MS**, MT
17	Do more research (look at scientific papers).	**Everyone**
18	Improve the tracker in order to increase counting performance.	**EF**
19	Evaluate the possibility of using Raspberry Pi machines.	**EF**, NW

5.4 Result

The most prominent result from this sprint was the realization that tracking and counting people waiting in line is extremely hard without any form of depth information. This realization, combined with the acquisition of a Microsoft Kinect sensor caused the focus of the project to shift entirely from the mono-camera approach to one using depth sensors. Due to a customer desire to run the system on a small, cheap computer a number of Raspberry Pi machines were recieved. Over 20 man-hours were spent on trying to make these communicate with the kinect without success. What we did finish however, was a kinect-based human segmentation, an improved tracker, stereo block matching and the collection of a 40 minute long kinect data set capturing several interesting scenarios.

5.4.1 Solutions we ended up using

The solutions described above that ended up being in use were the kinect-based human segmentation together with a new tracker.

5.4.2 What did we do well?

The kinect based image processing pipeline works well and provides promising results together with the new tracker.

5.4.3 What improvements are necessary?

Better communication is required when big changes occur, such as switching to depth-sensors or abandoning a platform (Raspberry PI). To solve this a new, non-digital SCRUM-board has been created on the whiteboard in the project room, and group members are encouraged to do their work in the project room.

Item	Result	People
1	Done.	**NW**, MS
2	Cancelled. Hours of tuning did not improve performance enough.	**NW**, MS
3	Done, however performance was not good enough to allow further development and tuning.	**GH**, EF
4	Done. Performance is not good enough.	**GH**, EF
5	Cancelled. Too complicated and too high computational complexity.	**GH**, EF
6	Done.	**GH**
7	Dropped, focus is shifted to the Kinect sensor.	**AS**, MT, MR
8	Evaluated but dropped due to block matching using OpenCV build-in functions being too slow.	**AS**, MT
9	Done using the Kinect sensor.	**AS**, MT, MR
10	Done.	**MS**, NW
11	Almost done.	**MR**, MS
12	Done.	**AS**, MT
13	In progress.	**MT**, All
14	Done.	**EF**
15	Done.	**MS**
16	Done.	**MS**, MT
17	Done.	Everyone
18	Done. The tracker now handles a lot more difficult cases.	**EF**
19	Done.	**GH**, NW

5.5 New Backlog Items

Several new backlog items were created as a result of this sprint, most of which are aimed towards finishing the problems, dealing with the final issues.

5.5.1 Stream live from the kinect sensor

Make the program run live using the kinect sensor on Windows as well as Mac OS X.

5.5.2 Evaluation data set

In order to make a somewhat reasonable estimation of system performance an evaluation data set needs to be collected from a different student kitchen.

6 Review of Sprint 4

Period: 2013-12-02 – 2013-12-13
Est. time: 330 hours, meetings included. Time spent: 365 hours, meetings included.

6.1 Sprint backlog

The main focus of this sprint is to put the entire system together, make a performance evaluation and create all the necessary documents. As much of the work is documentation, more work will take place in smaller groups, hence there are fewer people designated to specific items.

6.2 Backlog items from previous sprints

These are the items created in the previous sprints that are picked up in this one. Most of the new items were created during the previous sprint.

6.2.1 Sprint review collection

Provide a collection of sprint reviews that together with the spreadsheet describe the project work, result and planning. This item was created during the first sprint.

6.2.2 Make the Kinect stream live on Mac OS X

Make it possible for the system to use the kinect sensor as an input on Mac OS X.

6.2.3 Make the Kinect stream live on Windows

Make it possible for the system to use the kinect sensor as an input on Windows.

6.3 Sprint plan table

Item	Description	People
1	Create figures for the report/DOxygen code reference manual. E.g. intermediate algorithm steps and a nice visualization of system performance.	**Everyone**
2	Collect an evaluation data set	**Everyone**
3	Write a system overview for the webpage and DOxygen manual.	**Everyone**
4	Perform a complete system test on both platforms, running both live and from file.	**Everyone**
5	Create a manual for the debug GUI.	**Everyone**
6	Create a manual for the installation procedure.	**Everyone**
7	Revise the requirement on multiple kinects.	**Everyone**
8	Add content to the web page.	**Everyone**
9	Create a non-GUI version of the program.	**GH**
10	Improve the presentation of evaluation performance.	**Everyone**
11	Send info about the number of people in the room over the network.	**Everyone**
12	Add a Kalman tracker for velocity estimation to improve qeue detection	Everyone
13	Finalize queue detection.	**Everyone**
14	Put together queue information over time.	**Everyone**
15	Investigate the situation in the demo room.	**Everyone**
16	Make the kinect run live on OS X.	**Everyone**
17	Make the kinect run live on Windows.	**Everyone**
18	Make the system more robust vs different sensor placements.	**Everyone**
19	Go over the DOxygent comments and clean up the code.	**Everyone**
20	Add content to the technical report.	**Everyone**

6.4 Result

The results of this sprint can be considered nothing more than excellent, as everything that absolutely needed to be finished by the time of the deadline was finished by the time of the deadline. Only a few new features were added and most of the time was spent hardening the system.

6.4.1 Solutions we ended up using

We ended up creating a GUI-free version of the system designed to run on some form of embedded system. A nice calibration GUI was added to simplify system installation. Finally we ended up using libcurl as the network API, since it it is more light-weight than the Qt network module.

6.4.2 What did we do well?

The most important thing that went well was the fact that everything was finished before the deadline.

6.4.3 What improvements are necessary?

No improvements are really necessary before the final seminar and as long as everyone works as hard as they did in order to create the final version of the report and system, the final presentation is likely to be a success.

Item	Result	People
1	Done, we are not 100% happy with the data sets but they are good enough.	NW, EF, MT, AS
2	Done, three new data sets of various quality have been recorded. However som more would still be great.	MT, NW, MS, AS
3	Partly done, a short overview is added to DOxygen but none to the website.	MS
4	In progress: Perform a complete system test on both platforms, running both live and from file.	MT, NW
5	Done: Create a manual for the debug GUI.	AS
6	Done: Create a manual for the installation procedure.	AS, MT
7	Discarded, things work just fine. Revise the requirement on multiple kinects.	–
8	Not done, no content is on the webpage yet.	–
9	Done: Create a non-GUI version of the program.	GH
10	Done: Improve the presentation of evaluation performance.	EF, MR
11	Done: Information about room occupancy and queue is transmitted over the internet.	MS
12	Add a Kalman tracker for velocity estimation to improve qeue detection	MR, MT
13	Done: Finalize queue detection.	NW
14	Done: Put together queue information over time.	NW
15	Done: Investigate the situation in the demo room.	MT, AS
16	Done: Make the kinect run live on OS X.	NW, GH
17	Done: Make the kinect run live on Windows.	AS, MS
18	In progress: Make the system more robust vs different sensor placements.	**Everyone**
19	Go over the DOxygent comments and clean up the code.	**Everyone**
20	Add content to the technical report.	**Everyone**

6.5 New Backlog Items

No new backlog items are created since this is the final sprint of the project, however there are some more things that are in need of an overview before the project demo and presentation. These items are listed below.

6.5.1 New backlog item 1

Create a poster to present the work on the final seminar.

6.5.2 New backlog item 2

Create a presentation for the final seminar.

6.5.3 New backlog item 3

Create content for the web page.

6.5.4 New backlog item 4

Collect more data.

6.5.5 New backlog item 5

Prepare a live demo.