

1 Review of Sprint 1

Period: 2013-09-30 – 2013-11-04

Est. time: No estimate was made

1.1 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.

1.2 Backlog items from previous sprints

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

1.2.1 Algorithm interface

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

1.2.2 DOxygen comments

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

1.3 Sprint plan table

Item	Description	Est. time	People
1	Collect more test data, several data sets with varying difficulties using a single camera above a room 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

1.4 Result

The result from this sprint is good, as all of the items except for one was finished, providing a great framework for more advanced 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.

1.4.1 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 objects that have to fit certain criteria (currently lifetime) in order to be considered real objects.

1.4.2 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.

1.4.3 What improvements are necessary?

The most important improvement needed in the next sprint is how group members communicate when making design decisions. 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.

1.5 Sprint result table

Item	Result	Time Spent	People
1	Done, multiple data sets with different difficulties were collected.	8	AS, EF
2	Done, labeled with ground truth data for tracker performance evaluation.	8	AS
3	Done, code skeleton is completely finished.	10	GH, MT
4	Done, except for Object.hpp, because of dependency issues. Backlog item was created.	4	MT
5	Done, still only tested for reading from one single file.	6	MS
6	Done, however, the algorithm is much too slow and needs improvements.	5	EF, MT
7	Done.	6	EF, MT, MS
8	Done, with the exception of net flow estimation.	15	EF, MT
9	Done.	20	AS, NW
10	Done. (forward only)	10	AS, NW
11	Done.	3	AS, NW
12	Dismissed, deemed unnecessary at the moment	0	–
13	Done.	14	MR, MS
14	Done.	8	MS, MT

1.6 New Backlog Items

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

1.6.1 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.

1.6.2 Object class

Make sure the Object class meets the code standard.

1.6.3 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.

1.6.4 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.