Kitchen Occupation

People counting using depth sensors.

Department of Electrical Engineering, Linköping University

Objectives

1. Create a system to monitor room usage intensity, primarily focusing on student kitchens.
2. The system needs to be cheap and easy to install and maintain.
3. The system should provide real-time information about room usage intensity.

System

- Hardware setup
  - One or several Microsoft Kinect cameras and a computational device with Internet connection is required, as well as access to the power grid.
- Portability
  - Cross platform in the sense that we support most UNIX-like systems and Windows.

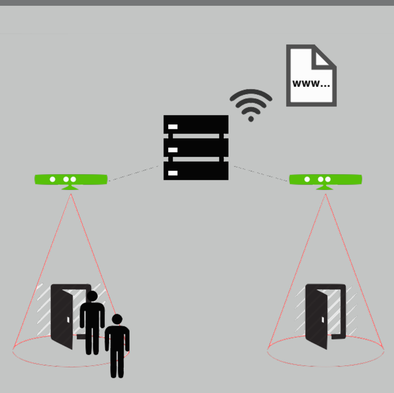


Figure: System Overview

Image Processing

- Human segmentation
  - The human segmentation is based on the assumption that human heads are distinguishable modes in the depth image and that people moving very close to each other seldom differ much more than a head in height (see figure). The later assumption seem to hold more often than one might think, as supported by over 5 hours of test data. The segmentation is realized by a series of threshold and morphological operations, Gaussian blurring, contour drawing and searches for local maxima's (see figure).

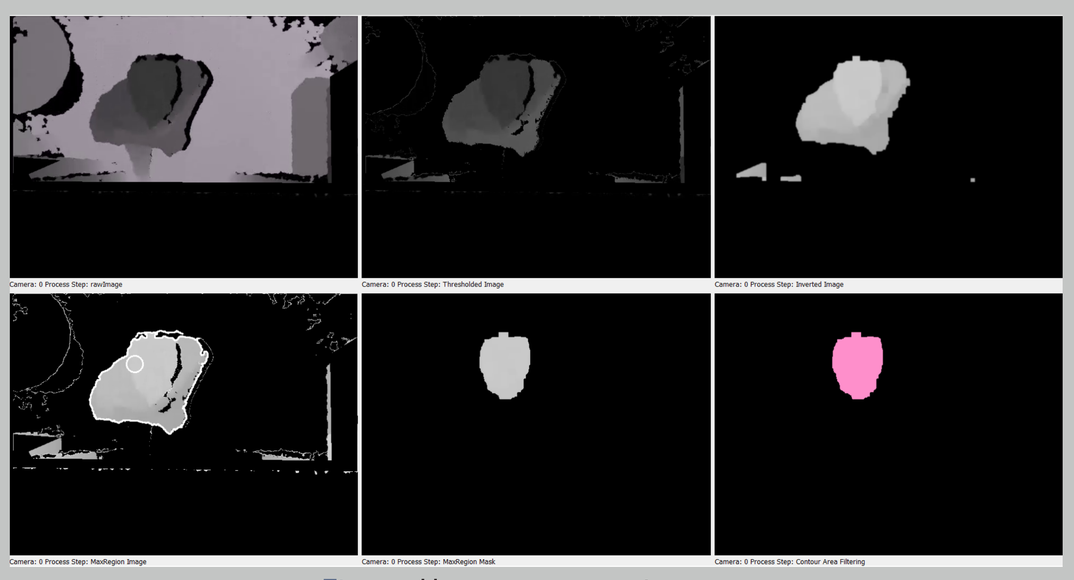
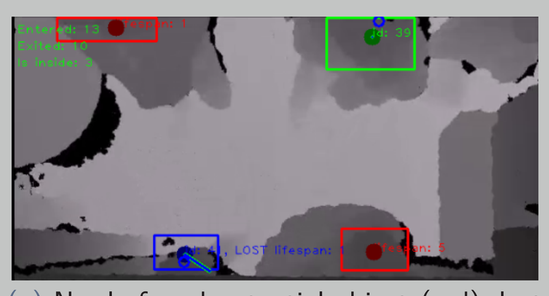


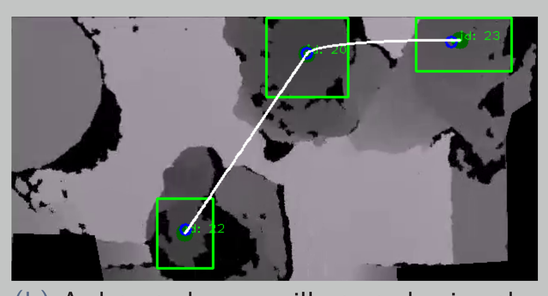
Figure: Human segmentation steps

Figure: Occlusion handling

- Tracking and counting.
  - The tracker pairs objects with each other from previous frame to the next. Pairs closest matching objects. Handles occlusion, outliers and noise.
  - Counting is done using user specified checkpoint lines and a door area.
- Queue detection.
  - Queues are detected using Bézier splines based on object velocities.
  - Queue severity is based on averaging detected queues over a few frames.



(a) Newly found potential objects(red), lost objects(blue) and objects(green)



(b) A detected queue, illustrated using the spline connecting the persons.

Software

- Built in C++ but very modular, extendable and configurable.
- Modular
  - Algorithms have a very clear general interface to the rest of the pipeline.
  - Multiple algorithms of the same type could be developed in parallel with minimal interface compliance effort.
  - An entire new computer vision approach could be switched to at the end of the project, entirely painless.
- Configurable
  - The entire pipeline can be reconfigured in the config file, including and rearranging algorithms as well as specifying their parameters without recompiling.
  - Possible to automate tests of hundreds of configurations, changing both algorithms and algorithm parameters.

Configuration & debugging GUI

- Transparent and flexible pipeline overview.
- Automatic online low-level profiling of every module.
- Configuration interface (see figure)
  - Doors (green)
  - Entering checkpoints (circles)
  - Exclusions (red)

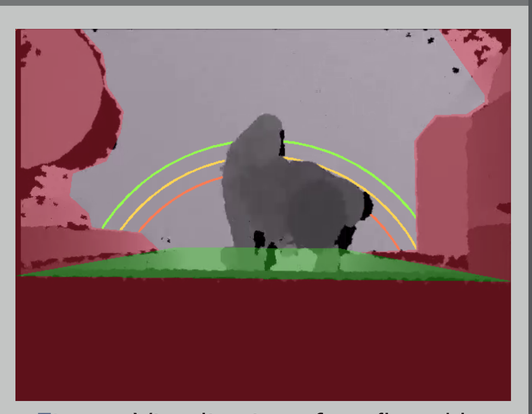


Figure: Visualization of configurables

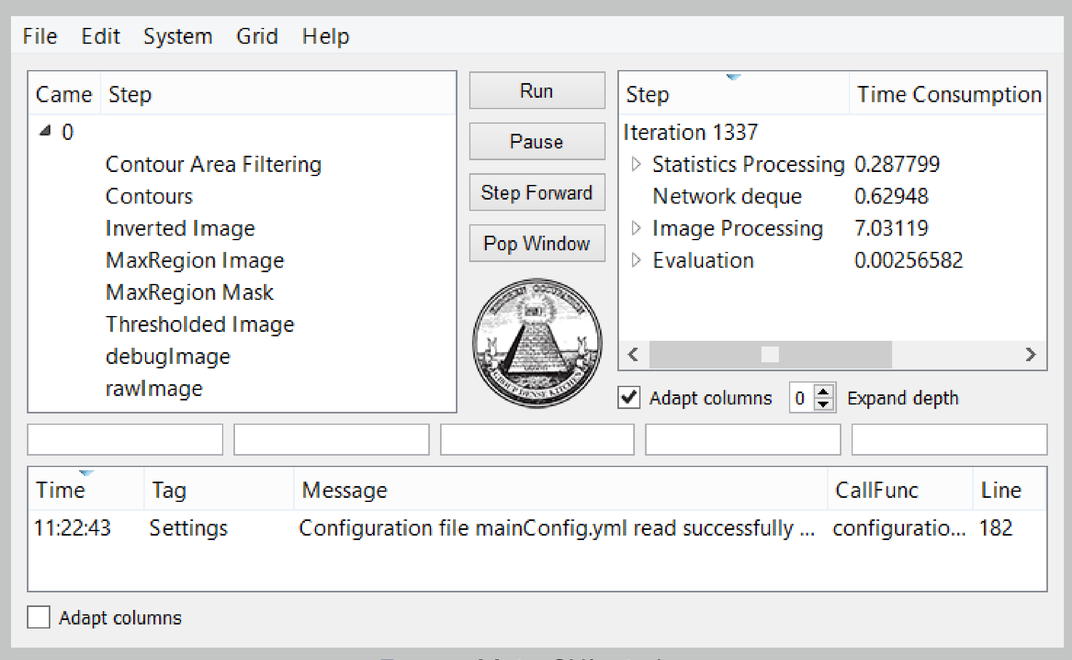


Figure: Main GUI window

Results: Table

- Final system performance

$$A_{in} = 1 - \left|\frac{\sum_{frames} in_{Est} - \sum_{frames} in_{GT}}{\sum_{frames} in_{GT}}\right| \quad (1)$$

$$A_{out} = 1 - \left|\frac{\sum_{frames} out_{Est} - \sum_{frames} out_{GT}}{\sum_{frames} out_{GT}}\right| \quad (2)$$

Sequence Name	Total entered (GT)	A_{in}	Total exited (GT)	A_{out}
Data seq. 1	108 (108) people	99 %	101 (104) people	97 %
Data seq. 2	122 (141) people	87 %	77 (91) people	85 %

Table: System performance in the two evaluation sequences

- Data seq. 1 & Data seq. 2 are two data sequences of 30 minutes each.

Conclusion

- The system provides high-precision real time people counting using the Microsoft Kinect sensor at low computational cost on a 2Ghz low-end CPU.
- The software architecture enables fast implementing and testing of different algorithms. It enable a very lightweight and high-performing vision pipeline.

Erik Fall Gustav Häger Malin Rudin Alexander Sjöholm Martin Svensson Mattias Tiger Nikolaus West