Kitchen Occupation

People-counting using depth sensors.

TSBB11 - Bilder och Grafik CDIO
Department of Electrical Engineering, Linköping University

Objectives

1. Create a system for monitoring 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 Properties

- Modular and easily configurable image processing pipeline.
- User-friendly debugging and configuration interfaces.
- Hardware consists of several Microsoft Kinect cameras and a computational device (figure 1).
- Cross-platform in the sense that most UNIX-like and Windows systems are supported.

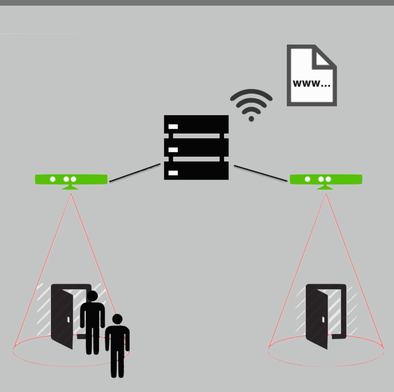

Figure 1: Rough system overview with main components.

Image Processing Pipeline

- Human segmentation
  - 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 radically in height (figure 3). It is achieved by a series of thresholdings, morphological operations, smoothings, contour drawings and searches for local maxima (figure 2).

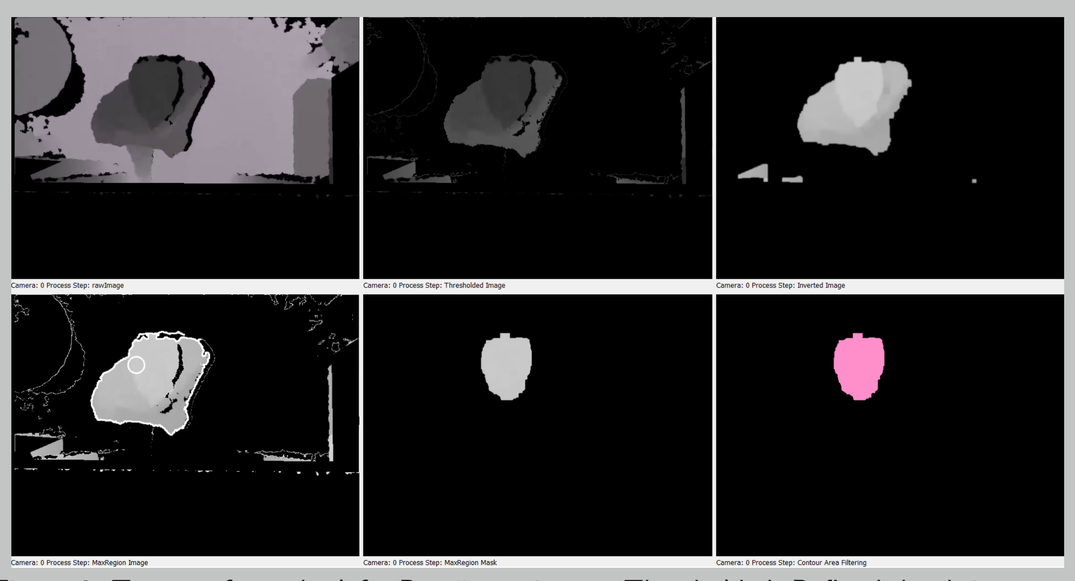

Figure 2: Top row from the left: Raw input image, Thresholded, Refined depth image. Bottom row from the left: Local Maxima detection, Head thresholding, Segmented heads.

Figure 3: From left: Raw imput image, Occlusion after thresholding, Successful segmentation

- Tracking and counting (figure 4)
  - Objects are paired with their closest matching object from the previous frame. Occlusion, outliers and noise are handled.
  - Counting is done using user-specified checkpoint lines and a door area.
- Queue detection (figure 5)
  - Objects connected by short Bézier curves, which has been fit to the objects' positions and velocities, are considered to be in a queue.
  - Queue severity is determined by queue detection frequency.

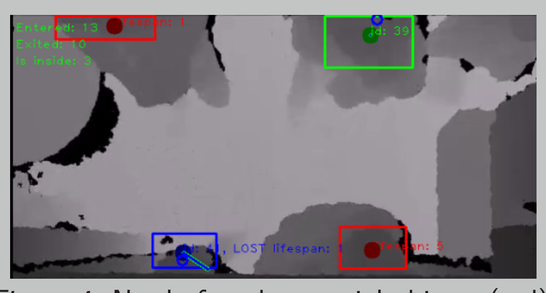

Figure 4: Newly found potential objects (red), lost objects (blue) and objects (green)

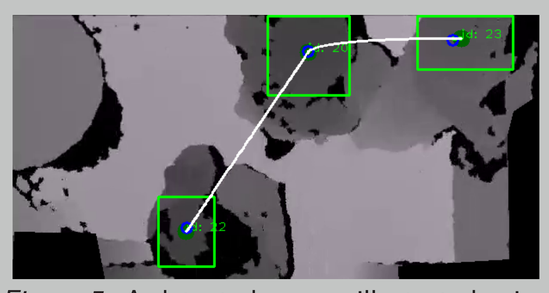

Figure 5: A detected queue, illustrated using the spline connecting the persons.

Software

- Built in C++ in a very modular, extendable and configurable fashion.
- Modular
  - Algorithms have a very clear general interface to the rest of the pipeline.
  - Multiple algorithms of the same type were developed in parallel, with minimal effort on interface compliance.
  - Allowed the entire computer vision approach to be switched near the end of the project with minimal effort.
- Configurable
  - The entire pipeline can be reconfigured from file. This includes addition, removal and rearrangement of algorithms, and parameter values.
  - It is possible to automate tests of hundreds of configurations, changing both algorithms and algorithm parameters without re-compiling.

Configuration & Debugging User Interfaces

- Transparent and flexible pipeline overview (figure 7)
- Automatic on-line low-level profiling of every system module (figure 7)
- Configuration interface (figure 6)
  - Doors (green)
  - Entering checkpoints (circles)
  - Excluded areas (red)

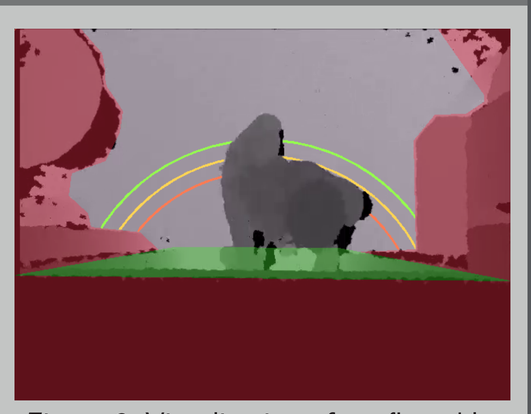

Figure 6: Visualization of configurables

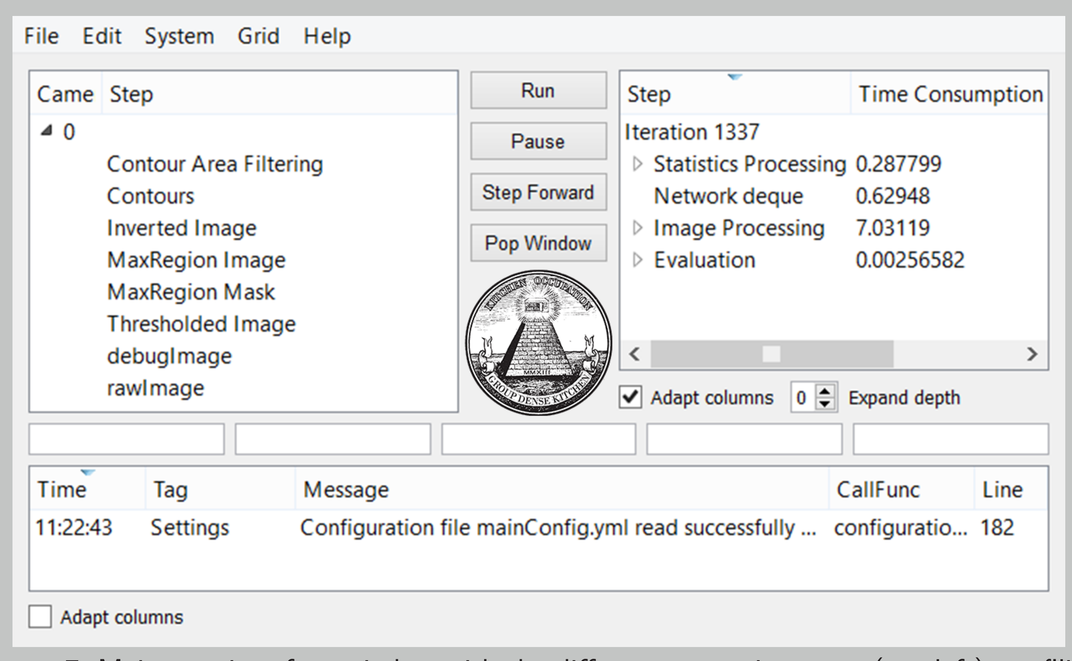

Figure 7: Main user interface window with the different processing steps (top left), profiling information for system modules (top right) and log messages (bottom)

Results

- System performance and equations used for evaluation.

Sequence Name	Total entered (GT)	A_{in}	Total exited (GT)	A_{out}	length
R-Kitchen	108 (108) people	100 %	101 (104) people	97 %	32 min
U-Kitchen	127 (122) people	96 %	136 (135) people	99 %	31 min
B25-Kitchen	131 (141) people	93 %	82 (91) people	90 %	30 min

System performance in the three evaluation sequences

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

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

Conclusion

- The system provides high-precision real-time people counting using the Microsoft Kinect sensor at low computational cost. (2GHz low-end CPU)
- Very simple assumptions are enough to get good results in practice.
- The software architecture enables fast implementing and testing of different algorithms. It allows for a lightweight and high-performing vision pipeline.

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