User Manual

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

List of Tables

Document history

Version	Date	Changes	Sign	Reviewed
0.1	2013–12–13	Initial draft	MS	MT

1 Installing the system

1.1 Hardware

Each Kinect camera must be installed above a door with no overlapping view shared with any other Kinect camera. The Kinect must point down or slightly angled towards the room. For optimal results the Kinect should be placed approximately 40 cm above the door to be able to detect tall even the tallest persons. Each Kinect must be connected to a power source, and to a device running the system software using USB.

1.2 Software

There are two versions of the software, one with a calibration and configuration GUI and one lightweight version without GUI. In order for the lightweight version to work a configuration file, presumably generated by the GUI version, is required. The configuration file is best generated using the configuration program, and then copied to the system running the non-GUI variant.

Linux, OS-X or Windows is required on the machine running the software. At least one Kinect camera must be connected before starting the software. More than one Kinect camear is currently only working on Linux and OS-X. Some software libraires are required to compile the program, these are listed in table 1.1.

Software	**Comments**
OpenCV2	Needed for general image processing
libFreenect	Needed for communication with kinect on unix like systems
OpenNI	Needed for communication with kinect on windows systems
libCurl	Needed to send http requests to the report API
QT5	Needed for the gui code, not used in headless variant

Table 1.1: Software libraries requried.

2 Calibrating the system

The system can be calibrated using the GUI for the most common calibration task. Using the config file all parameters are accessable for further calibration.

2.1 GUI

A threshold level is used to adjust the system for the current installation height of the camera. It sets a configuration parameter called lowestDistanceOverFloor. This is the limit of how short a person can be. The threshold should be set so that a "normal" person's chest is not removed by the thresholding.

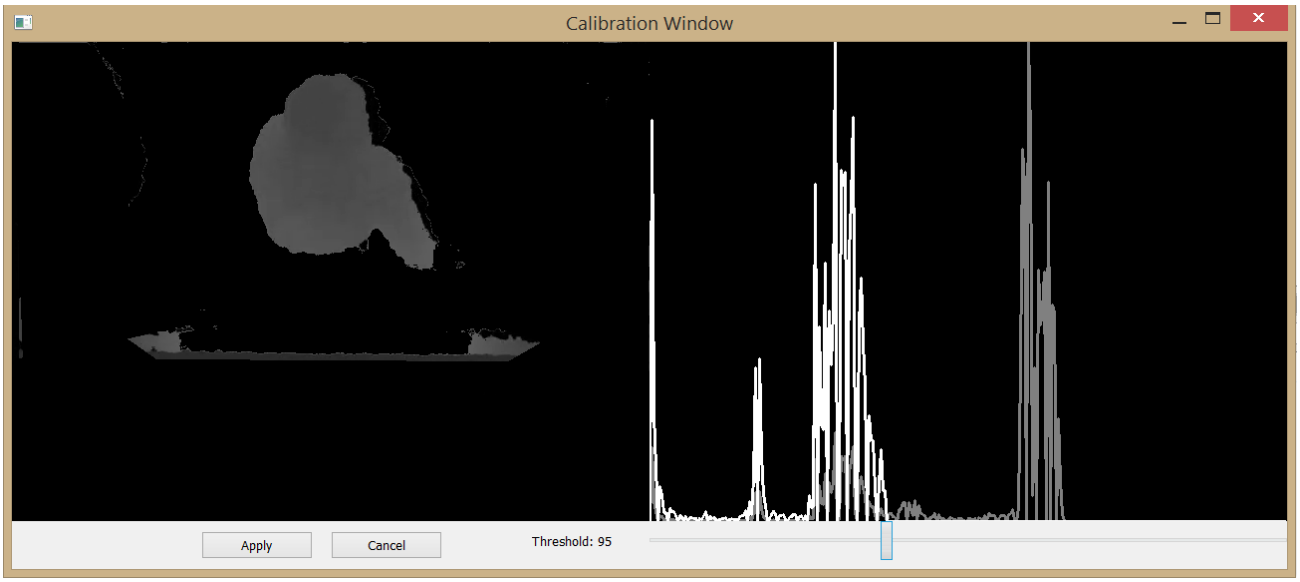

Figure 2.1: *Calibrating the lowestDistanceOverFloor threshold. A histogram i shown to help the user to se how much off different heights are present in the image. The selected heights are gray shaded.*

2.2 Config file

The config file used by the algorithm pipeline is called *dense_conf.yml*. In it any configurable variable in any algorithm currently selected to be in the pipeline can be specified. The pipeline itself can also be specified, allowing fo rapid swapping of algorithms. The most useful variables in the current pipeline are shown in table 2.1.

Variable	**Algorithm/Program-part**	**Description**
runFromFile	Network	If set to 1, the video sources are files found in the paths *videoFilePaths*.
videoFilePaths	Network	The paths to the video files used if running from file.
useKinect	Network	If set to 1, the video sources are from Microsof Kinect cameras.
TrackingMaximumDistance	Tracking	The maximum distance an object can be considered to have moved since last frame.
TrackingMinimzumLifeSpan	Tracking	The minimal time (in # frames) a potential object must have existed (and been tracked) before it is considered a real object.
TrackingMaximumTimeLost	Tracking	The maximum time (in # frames) an object is allowed to be lost before it is forgotten.
lowestDistanceOverFloor	Kinect Segmentation	The limit (height units) of how short a person can be. Set this variable using the GUI calibration utility described previously.
webServerUrl	Network	The address to the web service to which results are reported.

Table 2.1: The most useful and common variables in the current pipeline.

Currently the pipeline consists of two major algorithms: *ImageProcessor* and *Analytics*. These in turn have several sub-algorithms that are executed in the order specified in the config file. The current pipeline is structured in the following way:

ImageProcessor:
- *KinectSegmentation*
- *TrackingBruteForce*

Analytics:
- *EntryExitCounter*
- *FlowEstimator*
- *QueDetector*
- *QueSeverityEstimator*

Any algorithm registed in the system can be used as a subalgorithm for any other algorithm, writing in the config file in the same way as with *KinectSegmentation* being a sub algorithm to *ImageProcessor*. To get an empty algorithm placeholder any none-registerd algorithm name (or variable name) may be used, such as:

UnregisterdName:
- *KinectSegmentation*
- *...*

It can now be used as a sub-algorithm to another algorithm (or placeholder algorithm):

ImageProcessors:
- *UnregisterdName*
- *...*

A placeholder algorithm works by just passing through initialization and processing calls to its sub-algorithms.

Warning: If you do not know what your are doing, do not modify the algorithm pipeline. Some algorithms have requirements which must be provided by earlier algorithms, the system will not run if these are not met. See the code documentation for further details on requirements and effetcs of different algorithms.

3 Configuration the system

The system can be configured using the GUI. Available configuration settings is checkpoint circles, door mask area, exclusion mask and grayscale height threshold settings.

The circles should be placed so persons walking into the room inevitable will pass all three lines. They should also be more inside the room compared to the door mask area. A good placement is illustrated in figure 3.1. Note that the red, most inner circle, includes the upper corners of the door frame. A too small inner circle will cause people to miss it and will therefore not be detected.

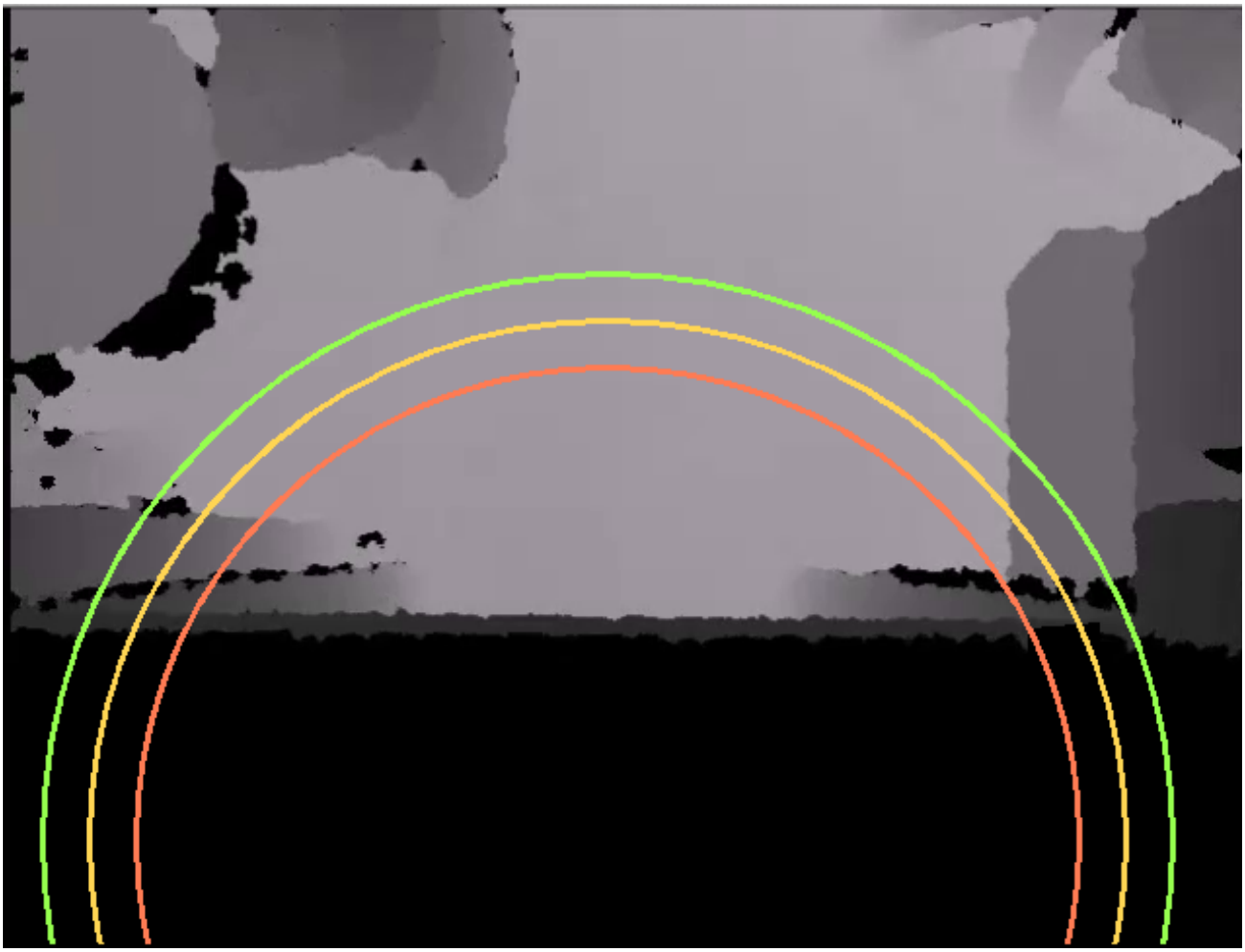

Figure 3.1: *A prefered placement of the circles.*

The door mask should cover the area close to the door where people appear. It is important to make this area big enough, rather too big than too small. It can, but should not cover the upper, most distant, part of the red circle, figure 3.2 illustrates this.

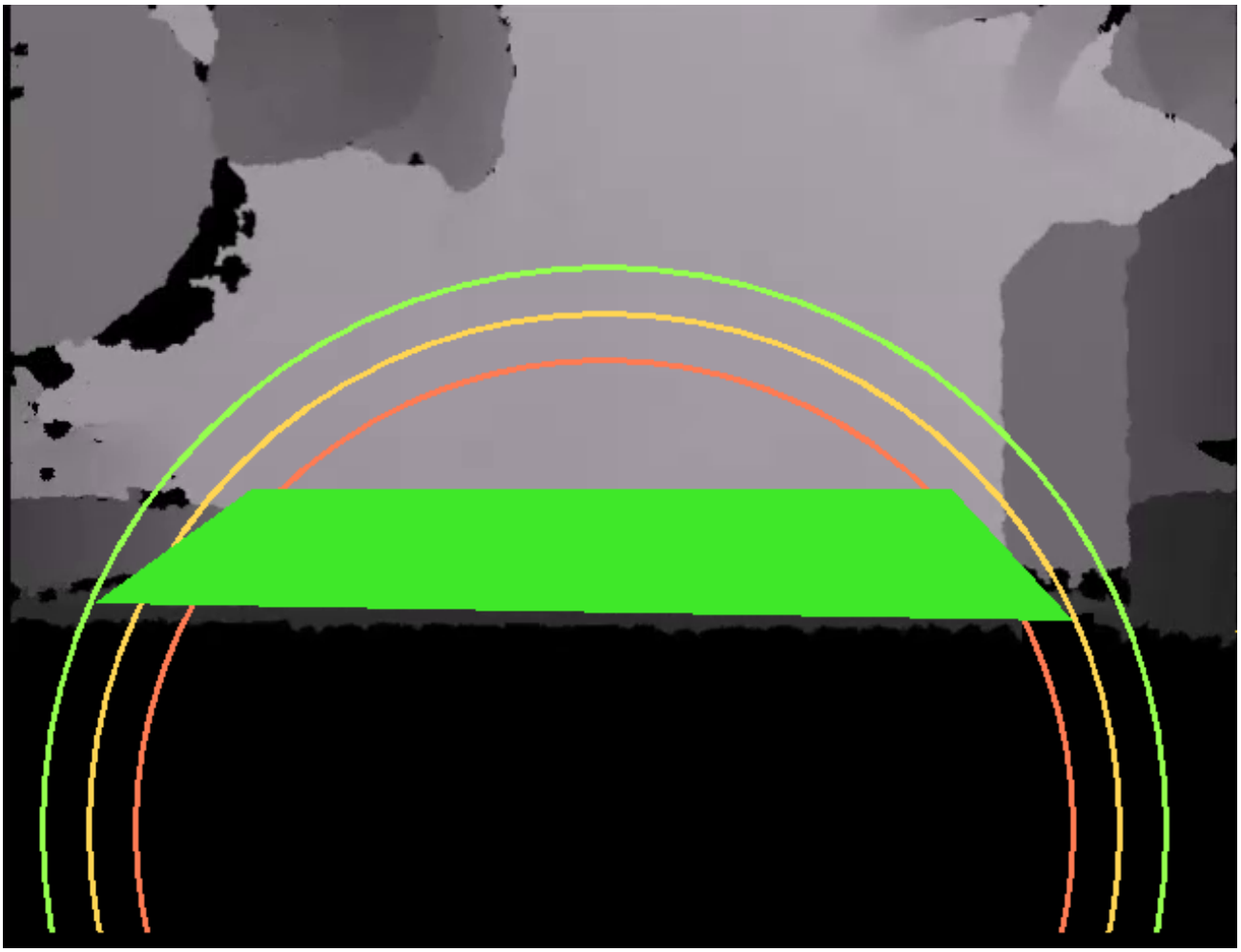

Figure 3.2: *The prefered placement of the door mask, the door mask is the green area.*

Exclusion masks should cover areas where people can not walk or appear. This could be areas like tables or areas behind the door (walls in this case), figure 3.3 illustrates this. Note that for long usage of the system, movable furniture should not be excluded.

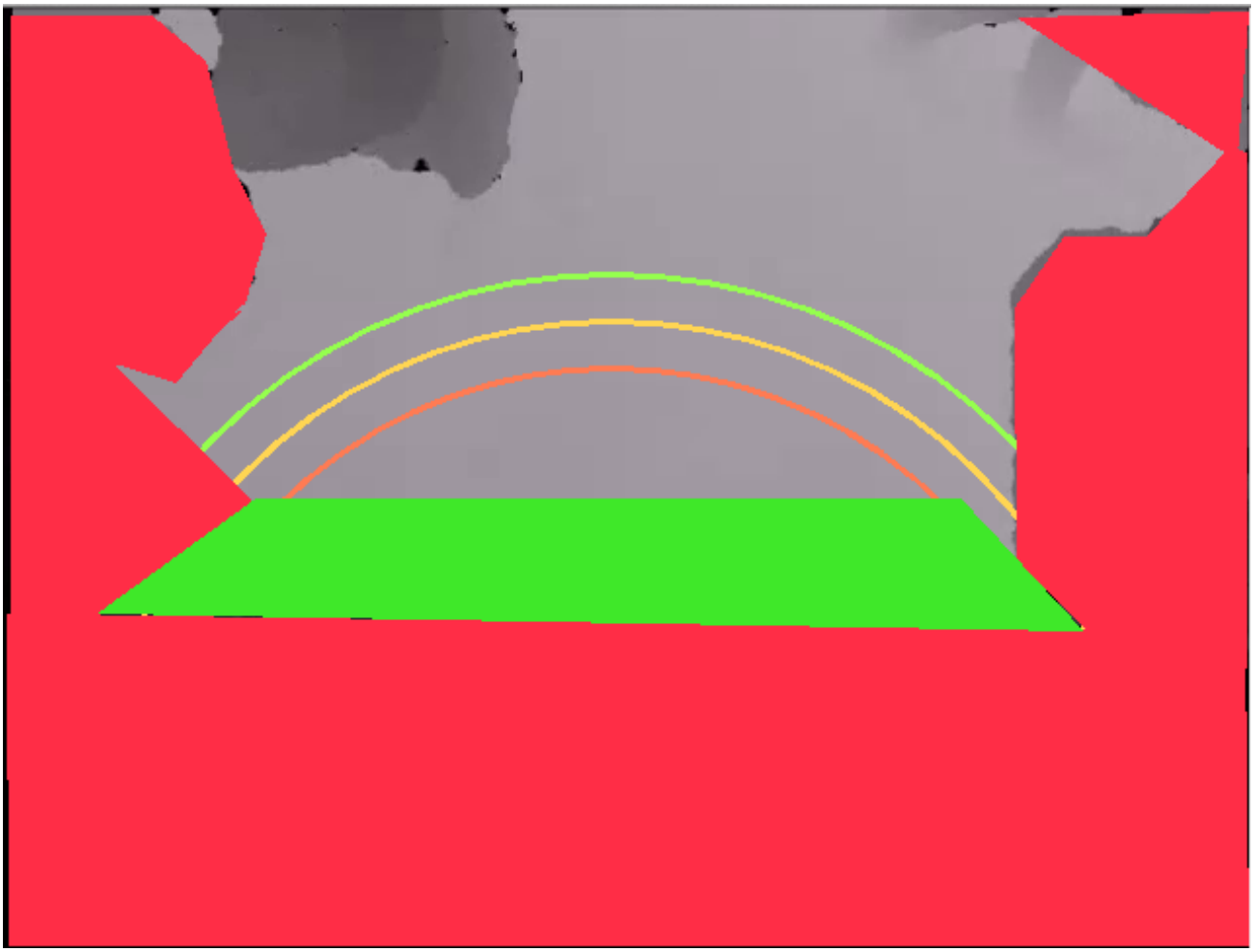

Figure 3.3: *Exclusion mask is marked as red. It covers areas where people can not walk or appear.*

References

[1] Gardel, A., Bravo, I., Jimenez, P., Lazaro, J.L. & Torquemada, A.
"*Statistical Background Models with Shadow Detection for Video Based Tracking*,"
Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on?? Page: 1-6.

[2] Zivkovic, Z. & Heijden, F.
"*Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction*,"
Pattern recognition letters, Vol. 27, No. 7. (2006), pp. 773-780.

[3] Bernardin, K. & Stiefelhagen, R (2008)
"*Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics*,"
Interactive Systems Lab, Institut für Theoretische Informatik,
Universität Karlsruhe, 76131 Karlsruhe, Germany

EXAMPLE REFERENCES ONLY, REMOVE BEFORE HANDING IN

[4] Sonka, M., Hlavac, V. & Boyle, R. *Image Processing, Analysis, and Machine Vision.*
Toronto: Thompson Learning, cop. 2008, 3rd ed., ISBN 0495244384.

[5] Wood, J. (2007) "*Statistical Background Models with Shadow Detection for Video Based Tracking*,"
Master thesis, Linköping University, Department of Electrical Engineering.

[6] Gustafsson, F., Ljung, L. & Millnert, M. *Signal Processing.*
Studentlitteratur, Lund, Sweden, 2011, 1st ed., ISBN 978–91–44–05835–1.

[7] "*CAVIAR: Context Aware Vision using Image-based Active Recognition*,"
EC Funded CAVIAR project/IST 2001 37540
http://homepages.inf.ed.ac.uk/rbf/CAVIAR/