Tommi Tikkanen

People detection and tracking using a network of low-cost depth cameras

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 20.1.2014

Thesis supervisor:

Prof. Arto Visala

Thesis advisor:

M.Sc. (Tech.) Otto Korkalo

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Tommi Tikkanen

Title: People detection and tracking using a network of low-cost depth cameras

Date: 20.1.2014 Language: English Number of pages:5+50

Department of Automation and Systems Technology

Professorship: Intelligent products Code: AS-84

Supervisor: Prof. Arto Visala

Advisor: M.Sc. (Tech.) Otto Korkalo

Automatic people detection is a widely adopted technology that has applications in retail stores, crowd management and surveillance. The goal of this work is to create a general purpose people detection framework. Studies on people detection, tracking and re-identification are reviewed. The emphasis is on people detection from depth images. Furthermore, an approach based on a network of smart depth cameras is presented. The performance is evaluated with four image sequences, totalling over 20 000 depth images. Experimental results show that simple and lightweight algorithms are very useful in practical applications.

Keywords: People Detection, People-flow counting, Kinect, Arbitrary oblique view cameras, Smart cameras

AALTO-YLIOPISTO
SÄHKOTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Tommi Tikkanen

Työn nimi: Ihmisvirtojen seuranta syvyyskameroilla

Päivämäärä: 20.1.2014 Kieli: Englanti Sivumäärä:5+50

Automaatio- ja systeemitekniikan laitos

Professuuri: Älykkäät tuotteet Koodi: AS-84

Valvoja: Prof. Arto Visala

Ohjaaja: DI Otto Korkalo

Automaattinen ihmisten havainnointi on jo laajalti käytetty teknologia, jolla on sovelluksia esimerkiksi kaupan ja turvallisuuden aloilla. Tämän diplomityön tarkoituksena on suunnitella yleiskäyttöinen järjestelmä ihmisten havainnointiin sisätiloissa. Tässä työssä ensin esitetään kirjallisuudesta löytyvät ratkaisut ihmisten havainnointiin, seurantaan ja tunnistamiseen. Painopiste on syvyyskuvaa hyödyntävissä havaitsemismenetelmissä. Lisäksi esittellään kehitetty älykkäiden syvyyskameroiden verkko. Havainnointitarkkuutta kokeillaan neljällä kuvasarjalla, jotka sisältävät yli 20 000 syvyyskuvaa. Tulokset ovat lupaavia ja näyttävät, että yksinkertaiset ja laskennallisesti kevyet ratkaisut sopivat hyvin käytännön sovelluksiin.

Avainsanat: Ihmisvirtojen laskeminen, ihmisen havaitseminen kuvasta, älykamerat, Kinect

Contents

Appendices

Abbreviations

AHPE	Asymmetry-based Histogram Plus Epitome
CPS	Custom Pictorial Structures
DSP	Digital Signal Processor
FPDW	Fastest Pedestrian Detector in the West
FPGA	Field-Programmable Gate Array
FPPI	False Positives Per Image
GMM	Gaussian Mixture Model
HDD	Histogram of Depth Difference
HOD	Histogram of Oriented Depths
HOG	Histogram of Oriented Gradients
INRIA	Institut national de recherche en informatique et en automatique (The French Institute for Research in Computer Science and Control)
IP	Internet Protocol
JPDA	Joint Probabilistic Data Association
LIDAR	Light Radar
MCMC	Markov Chain Monte Carlo
MHT	Multiple Hypothesis Tracker
MoG	Mixture of Gaussians
NNKF	Nearest-Neighbour Kalman Filter
RANSAC	Random Sample Consensus
RDSF	Relational Depth Similarity Features
RGB	Red Green Blue
RJMCMC	Reverse Jump Markov Chain Monte Carlo
ROI	Region Of Interest
SDALF	Symmetry-Driven Accumulation of Local Features
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
TOF	Time Of Flight
UDP/IP	User Datagram Protocol / Internet Protocol
ViBe	Visual Background Extractor

1 Introduction

People counting and tracking are technologies that are already widely being used on many fields of research and business. Applications include business intelligence in retail stores [1, 2, 3], surveillance [4], crowd management [5], transport [6], user interfaces [7] and intelligent environments [8].

Motivation for people tracking varies on each field. In retail stores, information about customer behaviour can be used to optimise shop layout to increase the sales, or to evaluate performance of an individual shop. In surveillance, for example activity in forbidden areas [9], violent behaviour or vandalism [10] can be detected. In crowd management, crowd behaviour can be monitored and predicted to avoid incidents such as the 2010 Love Parade disaster in Germany, where 21 people died and more than 500 were injured [11]. Furthermore, information about crowds and people flow indoors can be utilized to improve energy efficiency by optimising air conditioning, heating and lighting, or to develop emergency evacuation strategies [12].

The goal of this work is to create a general-purpose people tracking framework that can further be customised to specific applications. One example of an application is to provide measurements about people flow for a multi-agent building evacuation simulator. Traditionally, evacuation behaviour has been analysed manually from a video sequence. Automatic people tracking reduces the work load of analysing video, but also provides extra information to aid simulation development, such as height and speed of the persons. A further aim is to combine measurements from a sparse camera network with the simulation in real time, generating a model-based real-time approximation of people flow in a building.

A comprehensive study by Teixeira et al. [13] anticipates that future human-sensing systems are likely to consist of either massive amounts of low-cost binary sensors, mobile phone sensors or a smaller amount of cameras placed in key locations. In this work, the last one of these options is researched, that is, sparse camera networks. The advantage over dense camera networks is significantly lower cost with no dramatical decrease in understanding the people flow. Naturally, tracking performance is superior compared to a pure simulation that has no measurements. The purpose is achieve a relatively high performance with a relatively low cost.

People counting and tracking systems are widely available as commercial solutions. Common technologies used in these products include mono cameras, stereo cameras [14] and thermal imaging [15]. However, these sensors do not provide fully satisfactory results. Mono cameras are the most affordable, but require moderate image processing, are not the most accurate and have issues with varying illumination. Stereo cameras are able to produce depth information and therefore have possibility to be more accurate, but there are problems with finding features for every pixel. Thermal sensors are very robust for people counting, because they are immune to changing illumination and do not need background modelling, therefore very little image processing is needed. However, thermal sensors are slightly more expensive than color cameras, and reasonably priced sensors have a low resolution, possibly limiting the maximum detection area.

For the best visibility and less missed detections, the majority of people counting products are cameras that are placed in the ceiling, pointing downwards. However, this is not often the optimal set-up, if the detection area needs to be maximized. This is the case especially if the room is not very tall. Also, re-identification from top-view is much more difficult. [16]

On the contrary, a big part of academic research concentrates on detecting humans in color images from the side. Identification based on e.g. color histogram and silhouette is easy, but occlusions are a problem.

However, for a general-purpose solution it is not desirable to let the algorithm limit the how cameras can be installed, as the optimal camera position depends on the environment. Depth information has value when developing algorithms that are independent of the camera angle. During the past 10 years, many such methods have been researched. Plan-view projections [16], i.e. occupancy and height maps, are an example how to utilise 3D information efficiently. The development of depth cameras, such as Microsoft Kinect, has lead to the widespread use of depth data.

To the best of our knowledge, sparse camera networks based on depth cameras have not been previously implemented. Also, people tracking and counting with low-cost depth cameras has only been evaluated in research, where practical deployment is not considered, i.e. processing time and ease of calibration has not been emphasized.

The focus of this work is in making depth cameras field deployable. This means that constrained resources of an embedded computer limit the selection of algorithms. Instead of inventing a whole new approach, the goal is to describe a system that has practical value. To show the validity of the approach, four data sets of more than three thousand depth images are gathered. Test results are promising and show that the system is reasonably designed.

The thesis is structured as follows. First, both historically remarkable and state-of-the-art approaches to people detection and tracking are presented in the background-section. Second, the selection of hardware and algorithms are presented and justified. In the results-section, the accuracy of the people detection algorithm is presented. Finally, the significance of these results is discussed.

2 Background

In this chapter, a short history of algorithms for people detection, tracking and re-identification are presented, with focus on latest publications. Also, imaging and processing hardware are considered, as they not only generate a constraint for algorithm implementation, but enable new approaches.

The field of people detection and tracking is developing fast. During the past 10 years, there have been many breakthroughs in the research. First, Dalal and Triggs [17] introduced Histogram of Oriented Gradients (HOG) in 2005, creating a basis for quick development of appearance-based detection. Between 2005 and 2012, many improvements and extensions of HOG were invented, one of the most notable being The Fastest Pedestrian Detection in the West (FPDW) in 2010 [18]. Second, Primesense and Microsoft released the Kinect depth camera in 2010, making dense, high resolution depth imaging available at a low price. Third, the accuracy and computational cost of many decades old background subtraction has been improved by Barnich and Droogenbroeck [19] with Visual Background Extractor (ViBe) in 2011. Fourth, multiple-object tracking has developed dramatically. First, particle filters and multiple-hypothesis trackers [20] gained popularity in the early 2000s. Furthermore, Berclaz et al. [21] were able to find the global optimum in a 100 frame time window, while reaching nearly real-time performance on a PC.

Meanwhile, commercial people tracking and counting companies have began to sell massive amounts of sensors to practical applications. For example, Irisys who started only in 2001, has now sold over 100 000 thermal cameras for e.g. queue management.

2.1 Hardware

During the past decades, many types of sensors and algorithms have been applied to people detection, counting and tracking. However, majority of current people detection research concentrates on camera-based approaches, as they are very strong compared to others due to their relatively low price, high spatial resolution and the ability to provide multiple dimensions of information, including size, shape, colour and texture. [13]

Cameras are one of the most computationally complex methods of people detection. However, as the price of processing power continues to decrease, new computer vision methods become applicable. Also, algorithms that were too heavy in the past, can now be run with low-cost hardware in real-time. This enables the field of computer vision and its applications to evolve fast, both now and in the future. For this reason, the focus of this work is only on camera based people detection and tracking in this study. A survey about other sensors used for human-sensing can be found in [13].

Since the release of Microsoft Kinect [22] in 2010 it has raised a lot of interest in the field of computer vision. The reason is clear: Kinect provides high quality depth images (figure 1) at a lower price compared to previous technologies. Although Microsoft Kinect is meant to be a game controller, it has recently lead to an explosion of activity in both the research and amateur programming communities [13].

Depth information is necessary in computer vision applications, where the three-dimensional shape of objects needs to be observed. Previously, depth images could be obtained using LIDARs (Ligth Radar), Time-of-Flight (TOF) cameras and stereo cameras. While Kinect-like depth cameras cost 100 – 200 euros, LIDARs and TOF cameras are an order of magnitude more expensive. Also, LIDARs suffer from large size.

Stereo cameras on the other hand, are rather inexpensive themselves, but have other flaws. First, they are affected by changing illumination and cannot operate in the darkness. Another problem appears when observing large areas of similar color and little edges, as it may be difficult to find features [23]. For this reason, depth maps generated with stereo imaging are often not as dense the ones from Kinect. Third, developing a stereo based depth sensing system requires a significant amount of knowledge and work. Price of the commercial stereo cameras with embedded image processing capabilities is thousands of euros. Examples of such devices include Tyzx DeepSea G2 [24] and e-con Systems Capella [25].

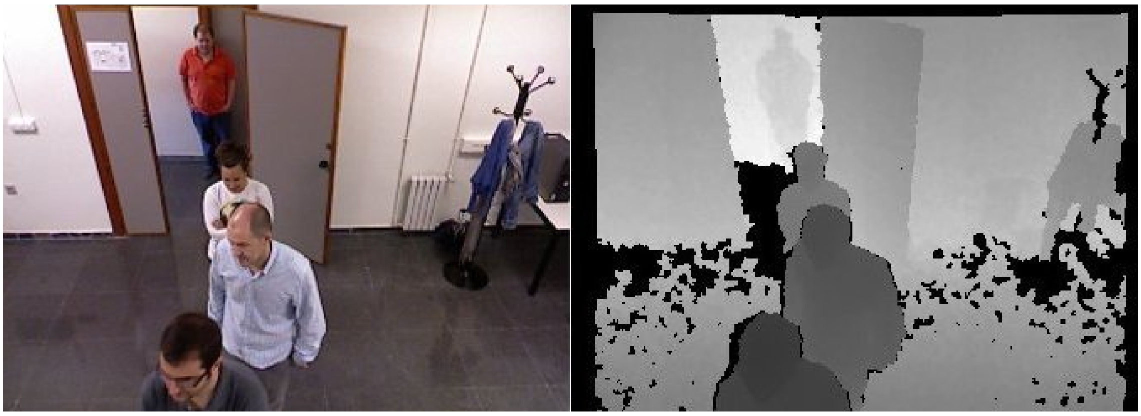


Figure 1: Depth image produced by Microsoft Kinect on the right side. Image from [26].

Sensor for the Kinect is designed by Primesense, which also sells the sensors for similar ASUS Xtion depth cameras. Primesense depth sensors are currently superior in 3D imaging for middle ranges (1 – 10 m), while LIDARs and stereo cameras still dominate outdoors and in longer ranges. In addition to generating depth maps, Primesense sensors also fuse depth data with color pixels on-board, resulting in a coloured 3D model to be easily available for any developer.

Kinect seems to be a superior depth camera in many occasions, performing fast and precise human sensing [13], but it still has two major limitations. First, the maximum range is 10 meters, which is less than what stereo cameras or LIDARs can achieve. Second, it cannot observe objects that are illuminated by direct sunlight, almost excluding use outdoors and disrupting use indoors near windows. For further information about the operational principle and applications of Kinect in machine vision, see [27].

2.2 People detection

People detection has many applications, such as human safety near robots, entertainment, surveillance and care for the elderly [18]. The large variety of application environments results in a large variety of different approaches. For example, one can consider three important fields: people counting, surveillance and pedestrian detection for cars. Pedestrian detection is constrained to horizontal camera angles. In people counting, a vertical angle is often preferred to avoid occlusions [28]. Finally, in many surveillance applications it is necessary to both see the faces of persons and to avoid occlusions, so camera is tilted downwards. Crowds tend to look very different from each point of view, therefore it is likely that the best combination of algorithms is different depending on the camera angle.

In this section, three relevant areas of people detection are introduced. First, different background models for background subtraction are presented. Background subtraction is widely used in surveillance and people counting applications. Second, appearance-based methods, popular for pedestrian detection, are reviewed. Although not as common in people counting, appearance-based methods are relevant, as they can be used to improve accuracy of people counting in combination with depth-based methods [29, 30]. Finally, the most relevant previous work is reviewed: people detection from depth images. Emphasis is on recent studies that employ Kinect-style sensors, but also earlier research with stereo- and time-of-flight cameras is presented.

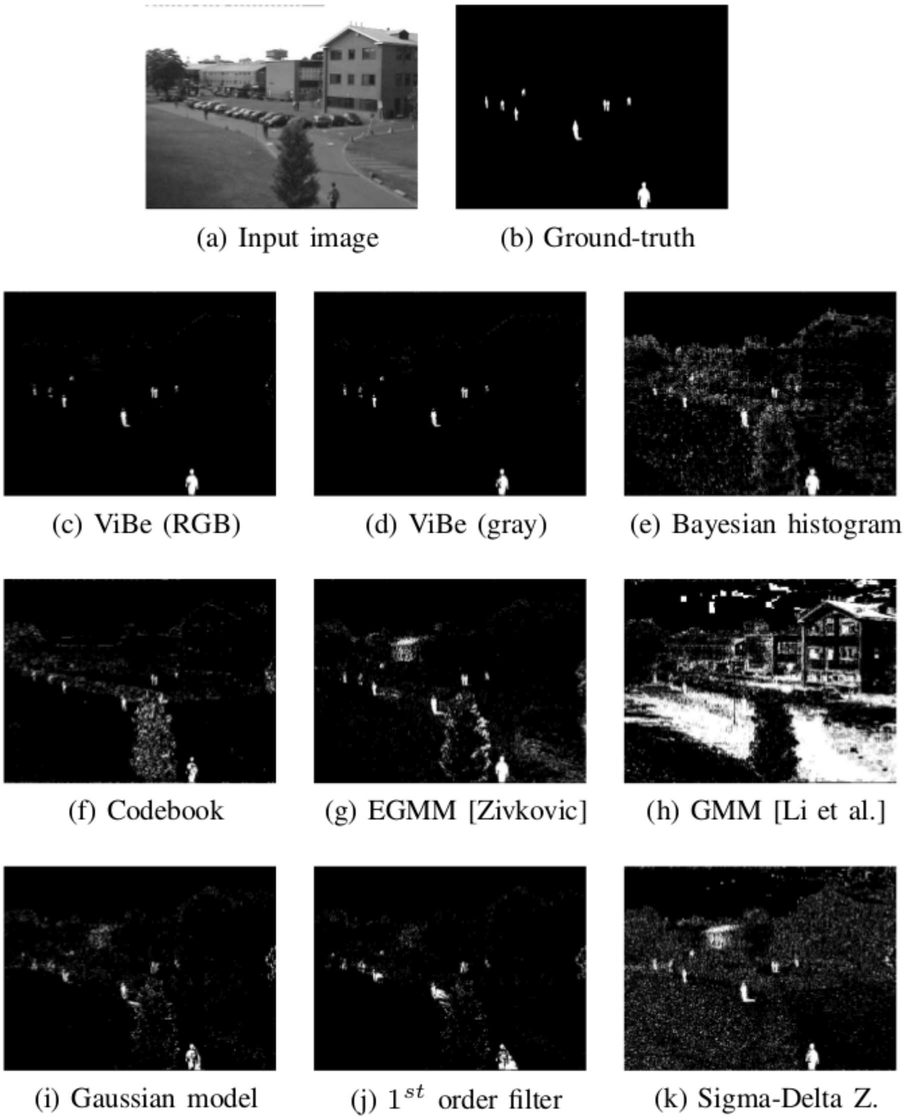
2.2.1 Background subtraction

The majority of currently deployed people detection approaches are based on extracting movement in the image by background subtraction. The reason for popularity is probably that it makes it fast to find objects of interest from the image [13]. It is suitable especially when the background scene is either static or slowly changing. However, there are major flaws in simple background subtraction [31]:

1. Natural oscillations in pixel intensity
2. Changes in lighting
3. Presence of repetitive background motion, such as waving tree leaves
4. Changes in position of static objects, such as furniture.

Many improvements for background modelling have been suggested, most popular being Gaussian Mixture Model (GMM) [32], also known as Mixture of Gaussians (MoG) [33]. The downside of GMM is in the assumptions it makes: that the background is more frequently visible than the foreground, and that foreground varies significantly less than background[19].

Lately, a simple algorithm called Visual Background Extractor (ViBe) [19] has surpassed GMM's performance both in terms of processing time and accuracy (figure 2). ViBe has been further improved by Droogenbroeck and Paquot [34], now called ViBe+.


(a) Input image

(b) Ground-truth

(c) ViBe (RGB)

(d) ViBe (gray)

(e) Bayesian histogram

(f) Codebook

(g) EGMM [Zivkovic]

(h) GMM [Li et al.]

(i) Gaussian model

(j) 1^{st} order filter

(k) Sigma-Delta Z.

Figure 2: ViBe surpasses the performance of GMM and various other background subtraction algorithms. Image from dataset PETS 2001. Labels are from [19], where different background models are explained.

With depth cameras background subtraction becomes a much more robust technique. The distortion from changing lighting and shadows are eliminated [27, 35]. Also, noise from the Primesense PS1080 depth sensor is relatively small, with standard deviation of less than 5 cm at 5 meter distance [36]. In indoor environments, the changes in furniture is perhaps the only challenge that depth-based background subtraction has to solve. Background models such as GMM or ViBe typically update themselves over time, adapting to changes in the background.

2.2.2 Pattern matching

Pattern matching is a very actively researched appearance-based technique for people detection. The advantage of appearance-based methods is that no background model is needed, so they are stable even in quickly evolving environments, such as on-board a driving car [13]. In the field of pedestrian detection, all the top performing methods are based on pattern matching [37]. Matching is usually done in one or many feature spaces, due to distortions caused by pose and illumination changes in image space matching [38]. Features are derived from the color image, for example using edge detection. Most often, pattern matching utilizes machine learning: the object's typical appearance is taught to the classifier by giving a large image database as a reference [13]. Various features for pattern matching have been proposed during the past years (table 1).

Table 1: Common features used in people detectors. Some of the features have been shown to be obsolete. Global chamfer distance is outperformed by local chamfer distance [39]. Also, Wojek and Schiele [40] conclude that HOG and Shape Context perform better than Haar-wavelet and Haar-like features.

Feature	**Author**
Global Chamfer distance	Barrow et al. [41]
Haar-wavelets	Papageorgiou et al. [42]
Haar-like features	Viola and Jones [43]
Scale-Invariant Feature Transform (SIFT)	Lowe [44], Lowe [45]
Histogram of Oriented Gradients (HOG)	Dalal and Triggs [17]
Shape Context	Mori et al. [38]
Local Chamfer Distance	Mori et al. [38]
Edgelets	Wu and Nevatia [46]
Shapelets	Sabzmeydani and Mori [47]

HOG and other versions of gradient histograms have quickly become the standard appearance-based people detector [37]. Therefore, other features are handled only briefly, concentrating on gradient histograms. A histogram of gradients is built in the following way. First, an image is divided into a grid of cells. Then, a histogram of the orientations of luminance gradients is computed in each cell. The histograms are normalized and concatenated into a single vector for the whole image. Different sizes of detection windows are slid across the histogram image. A linear Support Vector Machine (SVM) classifies the resulting vectors into person or non-person. [17]

After its introduction in 2005, many improvements for HOG have been developed. Dalal et al. [48] extended HOG to include the use of motion, decreasing the amount of false positives dramatically. Schwartz et al. [49] further incorporated texture information. Zhu et al. [50] improved performance by an order of magnitude using a cascade of rejectors, while keeping accuracy almost similar.

While no single feature has been shown to outperform HOG, additional features can provide complementary information. Wojek and Schiele [40] showed how a combination of Haar-like features, shapelets, shape context and HOG features outperforms any individual feature.

The accuracy of pattern matching has improved fast during the last decade. The Viola and Jones detector in 2001 [43] had approximately 10 false positives per image (FPPI) on the INRIA dataset [51] (80 % detection rate). In 2005, Dalal and Triggs decreased FPPI to 1, while the latest methods with more advanced learning methods and combinations of features have reached 0.1 FPPI on 80 % detection rate. [37]

Although the detection results for the latest methods might seem impressive, there are still many limitations that need to be overcome [18]. First, computational cost of many algorithms is notably high, processing less than 1 frame per second on a desktop PC [37]. The latest pedestrian detectors have partly solved this problem by selectively decreasing the number of iterations in sliding window matching. Such methods include VeryFast by Benenson et al. [52] and the Fastest Pedestrian Detector in the West (FPDW) [37]; these two run real-time on PC hardware, with similar or better recognition ability compared to their predecessors.

Second, HOG and other methods based on gradient histograms do not perform very well in case of occlusions [18]. Various ways to improve HOG in occluded scenes have been tried. Wang et al. [53] combined HOG with local binary patterns to handle partial occlusions. Salas and Tomasi [29] improved detection results by combining HOG with depth data from Kinect, further discussed in section 2.2.4. Other solutions to the occlusion problem are discussed in the section 2.2.3.

Finally, Dollár et al. [18] mention that low resolution, e.g. humans less than 50 pixels tall, is another serious challenge for even the best gradient histogram detectors. Besides solving problems with occlusions and small scales, they have listed five other areas of research that should be looked into, if these detectors want to be improved.

1. *Motion features.* The detector with the highest accuracy [54] in Dollár's comparison is the only one that takes advantage of motion. Dollár et al. [18] conclude that motion is a very effective method for human perception and thus a very promising research direction. However, the method in [54] is computationally very complex.

2. *Temporal two-way integration of tracker and detector.* Studies show that the probabilistic prediction of human location improves detection results.

3. *Ground-plane assumption.* Again, knowing where to look from improves results.

4. *Novel features.* The best detectors use multiple other features in combination with gradient histograms. It is expected that new independent features result in additional gains.

5. *Better datasets for testing.* Commonly, INRIA person dataset [51] is used to evaluate gradient histogram detectors [18]. The problem is that it contains

very few occlusions, and therefore it is too forgiving to algorithms that cannot cope with them.

2.2.3 Counting people in dense crowds

Since a major part of people detection research concentrates on pedestrians, the previously presented algorithms also favour these applications. It has to be kept in mind that pedestrian detection and people counting often have different priorities. In pedestrian detection, it is important to keep the frequency of missed detections very low, since a failure to detect a person can lead to injury or death [55]. In people flow monitoring applications, accurate people count and maintaining track of an individual are more important, therefore occlusion handling is crucial. As camera is positioned higher to avoid occlusions, more people become visible in a dense crowd. However, only the upper part of the body is visible of many individuals, e.g. head and shoulders.

While in sparse crowds, foreground segmentation combined with connected-component analysis or point clustering may be enough to locate individual objects, in dense crowd the problem of overlapping people becomes very significant (figure 3).

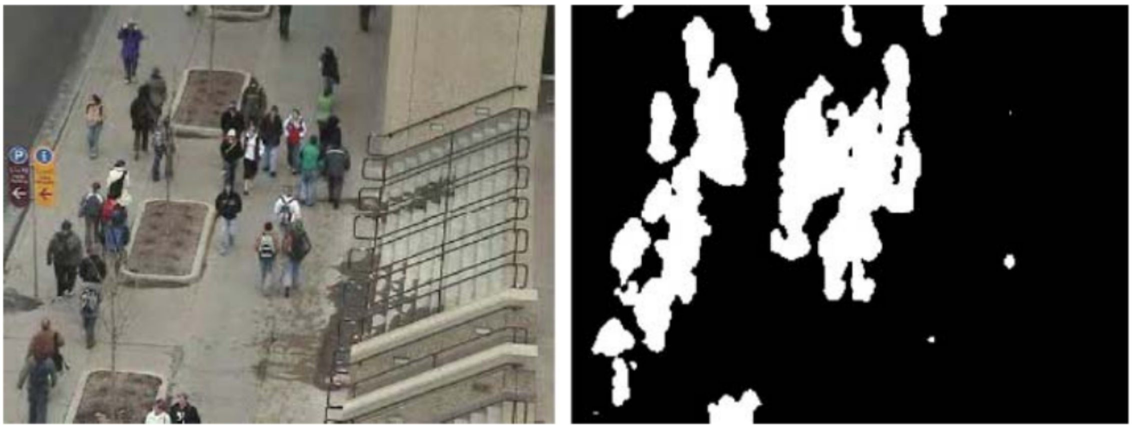


Figure 3: An example of a dense group where it is hard to separate foreground blobs for counting or locating the individuals. Image from [56].

In dense crowds, foreground segmentation is still useful, but needs to be combined with additional methods. Kilambi et al. [57] showed one way to handle big groups of e.g. 10 people after foreground extraction. The approach includes calculating the area of person's projection on the ground and after this, fitting ellipses on the group. Zhang et al. [58] have a similar method where they use a three-ellipse human shape model and additionally utilize color histograms to improve tracking. Zhao and Nevatia [59] combined four-ellipse human model segmentation with head detections from foreground and intensity image edges.

Rodriguez et al. [60] applied crowd density estimation to minimize the error of people count in a scene. The principle is to minimize the difference between density by density estimation and density by people detection. In other words, detections are added to places where they are too few, and removed from places with too many of them. This method implements one of the most accurate object detectors available [61], yet is able to significantly reduce false positives and increase true positives (figure 4).

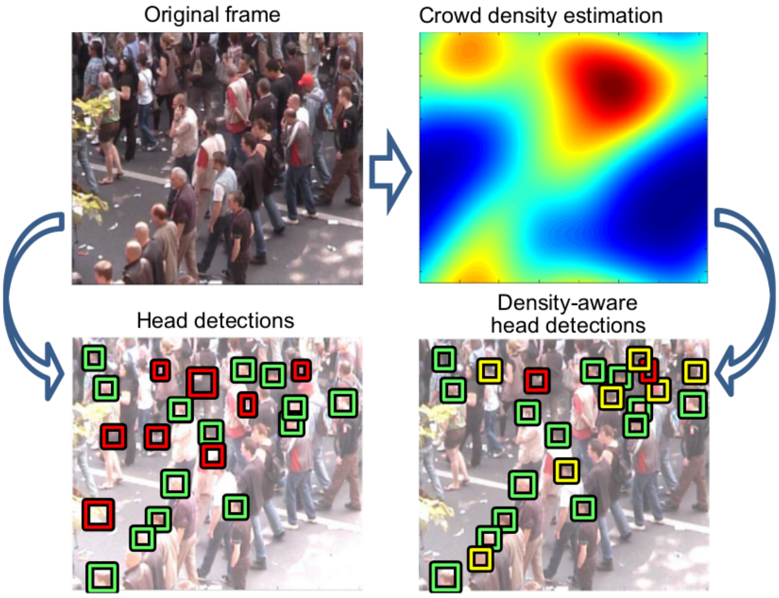


Figure 4: Crowd density estimation improves detection results. True positives are shown as green rectangles, false positives as red. Yellow rectangles are new true positives received with a crowd density estimate (top-right image). Image from [60].

2.2.4 Depth-based detection

According to Harville [16], depth data has great potential for improving people tracking performance for many reasons:

1. Depth is a powerful cue for foreground segmentation
2. Three-dimensional shape and metric size information improve foreground object classification, i.e. humans are better distinguished from other objects.
3. Occlusions can be detected and handled more explicitly.
4. New types of features for matching person descriptions across time become available, allowing better data association and re-identification.
5. A third dimension for prediction in tracking is provided.

The third dimension enables new algorithms for people detection, for example occupancy maps. On the other hand, many of the depth-based methods are just variations of color or intensity image algorithms. HOG, point clustering, connected-component analysis and background subtraction have been re-implemented using depth images or point clouds. Interest in people detection from depth images has increased with the development of processing and imaging hardware. Earliest depth-based people detectors were published in the 1990s, mostly working on stereo disparity images. In the late 2000s, novel time-of-flight (TOF) cameras gained popularity. Nowadays research is widely done with Kinect-style structured light sensors, but stereo- and TOF cameras are still actively used in research.

The first real-time stereo vision systems were published approximately in 1996, one of them by Kanade et al. [62]. Early depth-based people detectors, by Eveland et al. [63] and Darrell et al. [64] concentrate on extracting foreground from the depth image, while methods for finding occluded individuals in a group are rather simple. Darrell et al. [64] find gradients with a magnitude of more than 20 centimeters to separate different people in a connected foreground area.

Depth measurements provided by stereo cameras have a much greater noise compared to color values. Also, much of the depth image is unusable due to low visual texture [16]. To cope with inaccuracies, Beymer [65] presented the idea of occupancy maps for people detection in the year 2000. An occupancy map divides the ground plane X-Y into a set of vertical bins. Each 3D point is then accumulated in one of these bins by its X-Y coordinates (figure 5). Finally, humans can be found in the occupancy map in various ways, e.g. matching a 2D Gaussian model to the map [65].

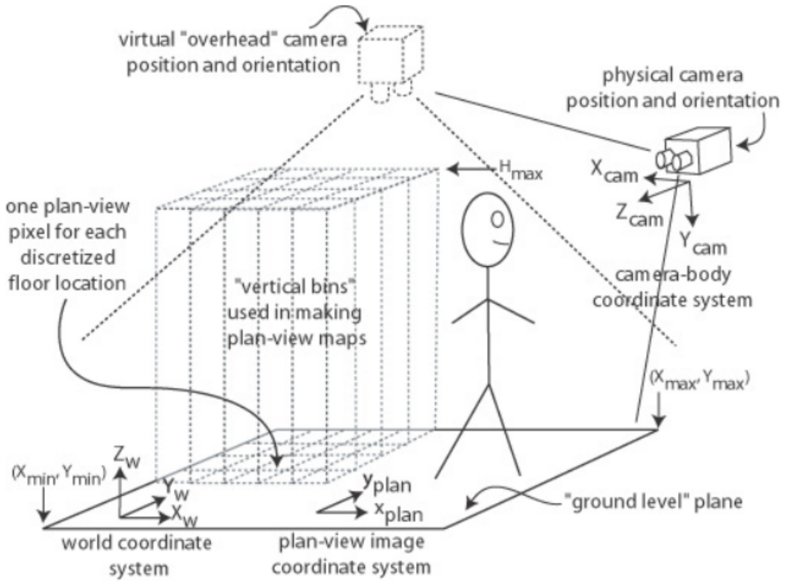

virtual "overhead" camera position and orientation
physical camera position and orientation
H_{max}
X_{cam}
Z_{cam}
Y_{cam}
camera-body coordinate system
one plan-view pixel for each discretized floor location
"vertical bins" used in making plan-view maps
(X_{max}, Y_{max})
(X_{min}, Y_{min})
Z_w
Y_w
X_w
y_{plan}
x_{plan}
"ground level" plane
world coordinate system
plan-view image coordinate system

Figure 5: Principle of transferring a camera to a virtual overhead pose for creating plan-view projections, such as occupancy and height maps. Image from [16].

However, Harville [16] notes that shape information in the vertical dimension is lost when using an occupancy map. Furthermore, partly occluded persons may be left unnoticed. A height map, on the other hand, preserves object shapes in the vertical dimension and handles occlusions better. However, height map alone would misinterpret small objects in the human head level. Harville [16] has solved the disadvantages of both methods by using occupancy statistics to refine the height map.

Nedevschi et al. [66] and Hordern and Kirchner [67] both used the occupancy map to extract regions of interest (ROI) from the depth image. Nedevschi et al. [66] further filtered the ROI by pattern matching side-view edge images. Finally, moving objects were counted as pedestrians if their motion signature, caused by moving limbs, was human-like. Hordern and Kirchner [67] projected 3D data onto 2D planes and used Fourier descriptors to classify the shapes (figure 6).

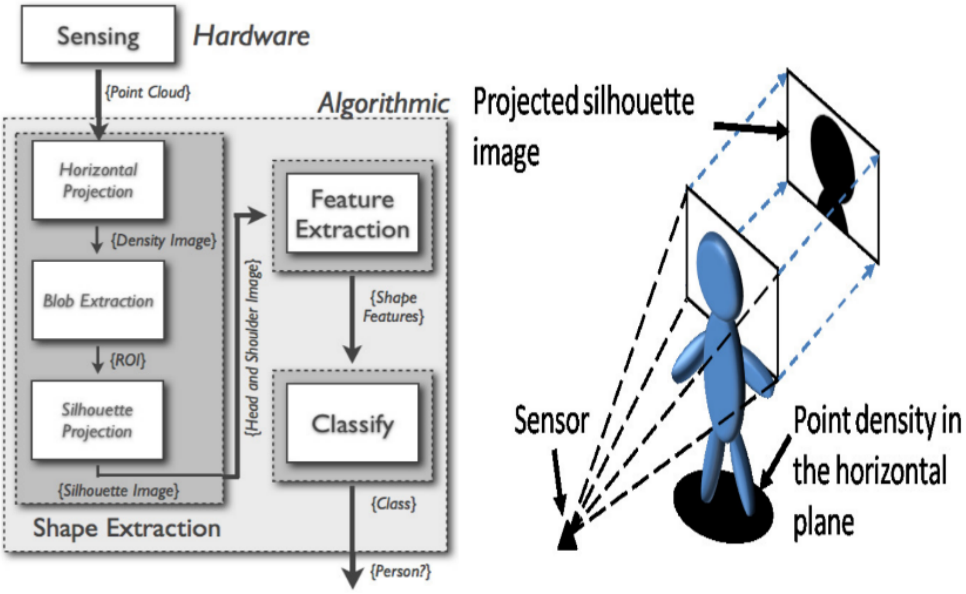


Figure 6: An example of using 2D projections of 3D data to detect humans. Occupancy map (i.e. density image) is used to extract foreground objects. Silhouette projection is used for object classification. Image from [67].

Among the first pattern matching attempts on depth images, Beymer [65] compared human shape template on stereo disparity image's foreground objects. Similarly, Luo and Guo [68] applied head-shoulder contour matching. Zhao and Thorpe [35] used a neural network to classify between human and non-human foreground objects. The classifier was taught with intensity gradient images. Satake and Miura [69] improved silhouette matching by overlapping silhouette templates, meaning that templates included multiple persons, possibly giving more accurate results in crowded scenes (figure 7). As noted in section 2.2.2, pattern matching such as HOG suffers from bad performance; various sizes of human feature templates have to be slid over the image, because distance at different parts of image is unknown. However, this is not the case with depth-based techniques, i.e. it is known what size of template needs to be applied at each point, decreasing iterations and making pattern matching a much lighter algorithm [70, 29, 30]. Another advantage of depth images is that foreground can be extracted much more reliably. Therefore, background is

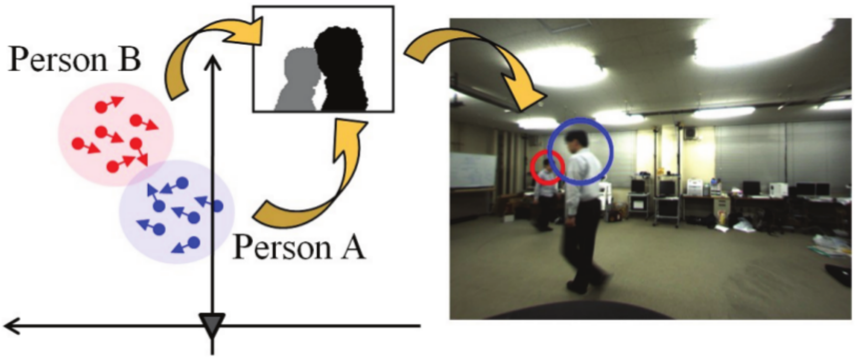


Figure 7: Compared to single person templates, overlapping silhouette templates are better suited to analyze crowds. Image from [69].

removed before applying pattern matching in almost all the papers evaluated for this study that employed Kinect.

Lately implemented algorithms have been strongly influenced by HOG. First, Ikemura and Fujiyoshi [70] presented Relational Depth Similarity Features (RDSF). It is a normalized depth value histogram of a small image patch (figure 8). Authors claim RDSF is a better feature than HOG, however this has not been verified by other researchers.

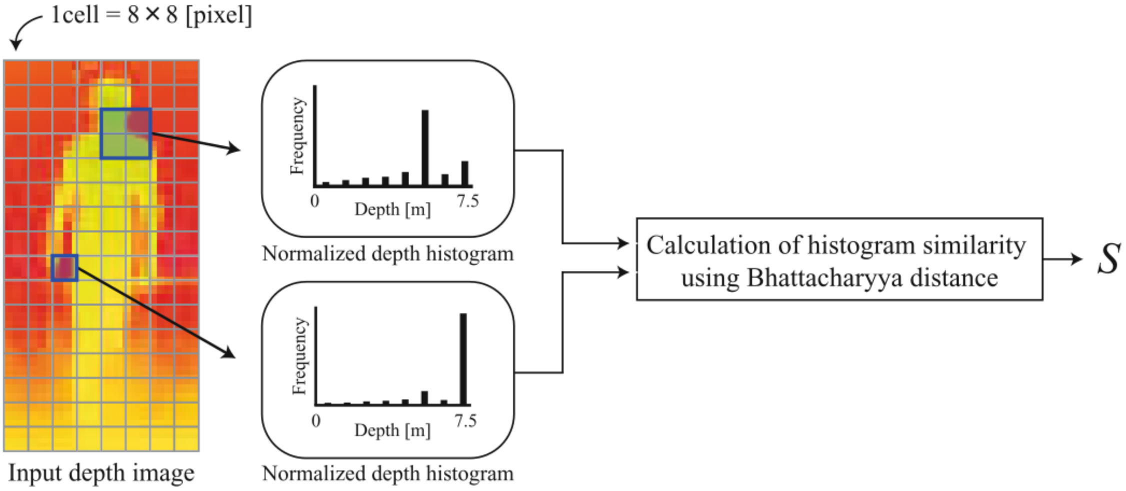

1cell = 8 × 8 [pixel]
Frequency
0
Depth [m]
7.5
Normalized depth histogram
Frequency
0
Depth [m]
7.5
Normalized depth histogram
Input depth image
Calculation of histogram similarity using Bhattacharyya distance
S

Figure 8: Relational Depth Similarity Features describe the normalized depth value distribution of a certain cell in an image patch. Image from [70].

One of the most popular depth image features has been Histogram of Depth Difference (HDD) [71], which is essentially similar to HOG, but with depth images. Simultaneously, Spinello and Arras [30] came up with a similar approach and called it Histogram of Oriented Depths (HOD). Furthermore, they presented Combo-HOD – a probabilistic method to combine results of HOG and HOD. Obviously, Combo-HOD is more accurate than either of the features independently. Also, if depth data is unavailable in case of sunlight or reflections, results from color image are still usable. Usually, HOG is applied in camera angles where human silhouette can be clearly seen. Tian et al. [72] have shown that HOD is effective also from top view.

While pattern matching is still mainly done in image space, some experiments on 3D shape matching have been conducted. First, Bajracharya et al. [73] measured various 3D properties of the foreground objects found with a polar-perspective occupancy map. These features include properties of the total point cloud, such as variance, and point counts of certain pre-set volumes, similar to a low resolution 3D histogram. Xia et al. [74] tried to find human heads by matching a 3D model of a hemisphere to a point cloud obtained by Kinect. The reason for selecting a hemisphere was view-invariance, as the human head will appear almost similar whether it is observed from the front, side, back or above. 3D model matching is used to remove false positives from 2D chamfer distance matching (figure 9).

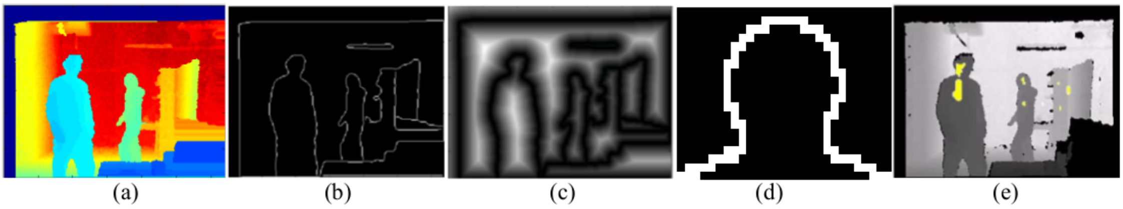


Figure 9: Xia et al. [74] used chamfer distance image with head-shoulder contour matching to locate potential heads. Furthermore, head candidates were filtered by matching a hemisphere model on their locations. a) Original depth image, b) edge image, c) chamfer distance image, d) head contour model, e) results of detection as yellow pixels. Image from [74].

If pattern matching is not used, almost always the foreground objects are extracted with background subtraction. In the evaluated 21 publications, only Munaro et al. [75] did not subtract the whole background – the floor was still detected and removed. Therefore, only two steps for people detection remain. First, it must be found which parts of the foreground correspond to which individual objects. Second, objects need to be classified as human or non-human. However, the second step is not even needed in many environments, where there are very few moving objects besides people.

One of the simplest ways to separate individuals from the foreground is to see which pixels of a height-thresholded foreground image are connected to each other, also called connected-component analysis. Hernandez-Lopez et al. [76] did this using an overhead Kinect, so that people are rarely overlapping in the image. Another approach, implemented for Kinect by Hsieh et al. [77], is to obtain a point cloud by a tilted camera and then rotate it to an overhead point of view, similarly as presented in figure 5.

The previously described method will obviously fail if people get too close to each other. For example limbs in the same planar coordinates erroneously connect two individuals as one detection, even if they are on a whole different height. This is a strong reason to use connected-component analysis in 3 dimensions if high quality 3D data is available, as Salas and Tomasi [29] have done. Even so, 3D connected-component analysis is not very robust in case people are actually touching. Therefore, Salas and Tomasi [29] used the point cloud connectivity to create initial track candidates called tracklets, which are further refined using HOG for corresponding color images. As an interesting detail, they use background subtraction on the occupancy grid instead of the depth image.

Seer et al. [78] used a more sophisticated method called complete-linkage clustering [79], which is more tolerant to weak links between individuals. This way, persons may be touching, e.g. holding hands, and still be recognized as two. Fu et al. [80] took advantage of convex hull segmentation to handle dense groups of people (figure 10). After finding best candidates with head-shoulder template matching, they filled the pedestrian in depth image in a way that adjacent points belong to the same segment if depth difference between them is below a certain threshold. Finally, unintentionally merged areas were split if the depth map inside the convex polygon had a strongly concave feature, i.e. a low enough valley between two heads.

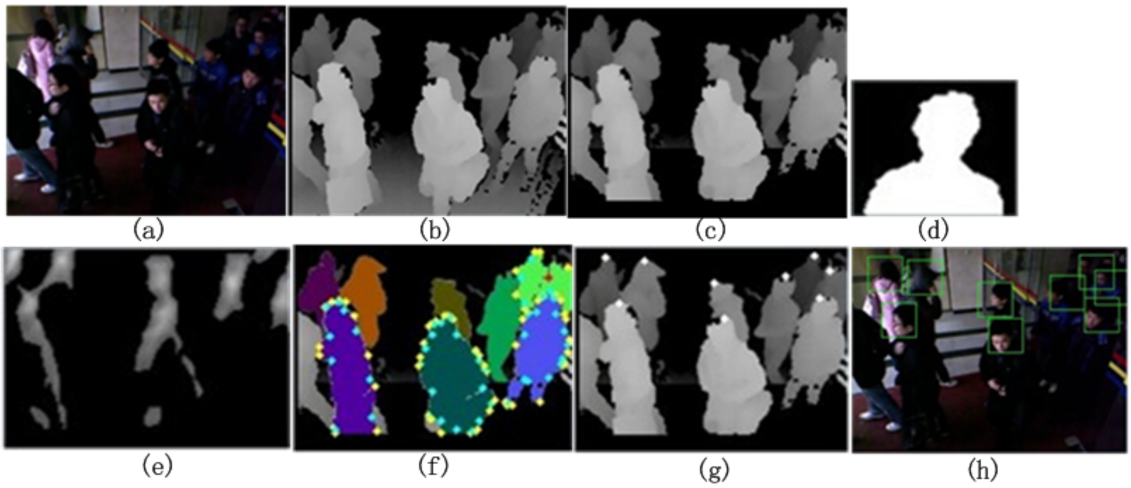
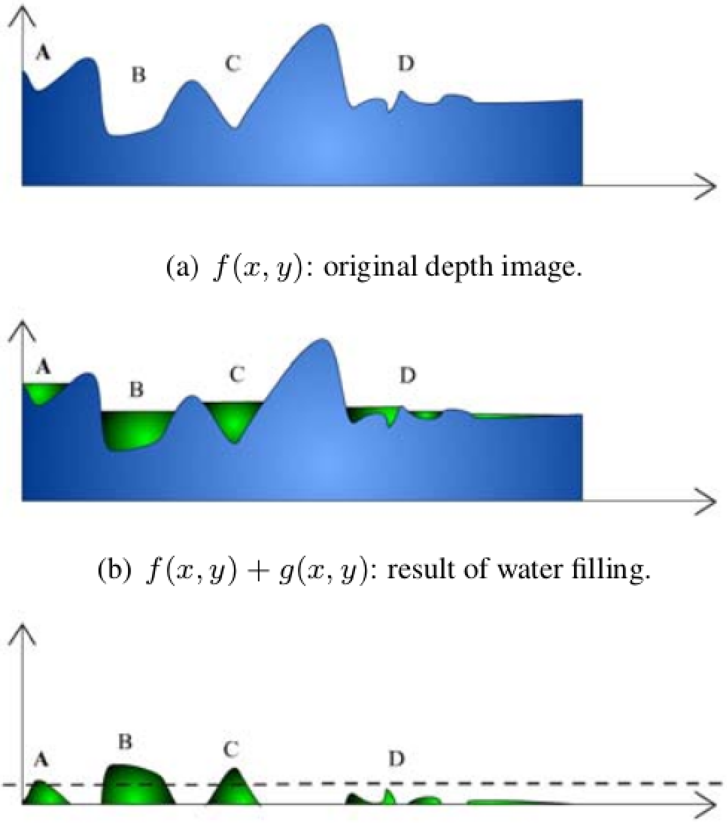


Figure 10: A depth-based approach by Fu et al. [80]. a) RGB image of the scene, b) depth image, c) depth foreground, d) head-shoulder pattern that is matched to the depth image, e) pattern matching result, f) convex hull segmentation result, g) heads found, h) bounding boxes for the heads found. Image from [80].

Zhang et al. [81] claim that pattern matching has two unsolved problems: multiple detections per person and missing detections by occlusion. As the head is normally the uppermost part of the human body, finding local maxima in a height map will give the positions of the heads. However, this method is not very robust if noise is present or if adjacent humans have very different heights. For example if a child is next to an adult, the adult's shoulder is often taller than the child's head. Zhang et al. [81] solve these problems with a method called water filling. An inverse height map is used as a surface where raindrops are randomly dropped on. Upon reaching the surface, the raindrop will move towards a lower point if one is found in the nearby environment. This is repeated until the raindrop cannot move any lower and the amount of water in the resting position is incremented. After dropping a large amount of water, humans are found by locating the bodies of water that exceed a certain deepness threshold (figure 11). The cleverness of water filling is that local maxima are not found using a constant size search window, but adapting to find the relevant maxima regardless if they are the actual maxima of a small or a larger area. Zhang et al. [81] have implemented the algorithm for an overhead camera, which is assumably the optimal set-up. However, it is unclear if the same approach could be used for tilted angles to cover a wider scene.



(a) $f(x, y)$: original depth image.

(b) $f(x, y) + g(x, y)$: result of water filling.

(c) $g(x, y)$: measure function

Figure 11: Water filling robustly finds head locations by accumulating rain drops, then finding the deepest ponds (green). Image from [81].

Additionally, to emphasize the most relevant research, publications where Kinect or a similar sensor was used, are listed with more details in table 2. As a conclusion, detection results are impressive, but there are some limitations. Silhouette matching methods are heavily dependent on the viewing angle and inevitably suffer from occlusions. Gradient histograms are able to function from any camera position, but require wide learning material from multiple angles. Overhead set-ups may cover a smaller area than side view or tilted cameras if the ceiling is low. It is unclear how well clustering or water filling works for other camera angles. Perhaps the most significant flaw in the evaluated publications is that practical usability is not considered enough: all detection systems run on PC hardware, which is powerconsuming and also increases the cost of processing hardware. On a real use case where cameras need to be deployed around a building, processing is rather performed on the camera node, to avoid network load by not sending image data through network.

Table 2: Previous applications of Kinect-like depth cameras in people detection and tracking.

Author	**Algorithms**	**Camera pose**
Salas and Tomasi [29]	HOG, occupancy map, 3D connected-component analysis	Side
Spinello and Arras [30]	HOG, HOD	Side
Xia et al. [74]	2D chamfer distance matching, 3D head model fitting	Side
Tian et al. [72]	HOG	Top
Seer et al. [78]	Complete-linkage clustering	Top
Zhang et al. [81]	Water filling	Top
Hernandez-Lopez et al. [76]	2D connected-component analysis	Top
Hsieh et al. [77]	2D connected-component analysis	Oblique
Fu et al. [80]	Head-shoulder template matching, connected-component analysis, convex hull segmentation	Oblique

Finally, to give a good overview about previous research on depth based people detection, the combinations of algorithms used in all the previously mentioned publications is shown in figure 12.

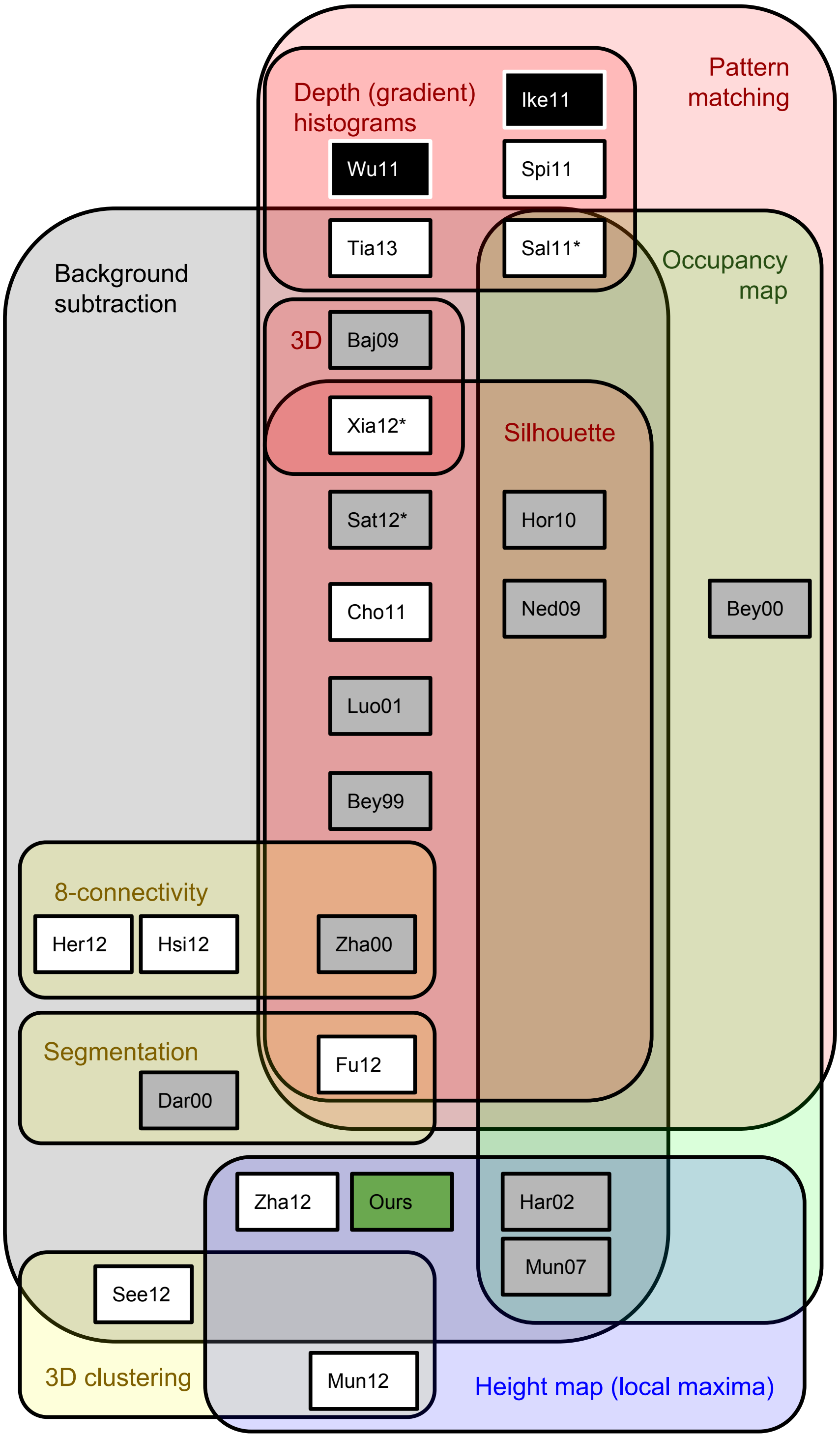


Figure 12: Selection of algorithms for people detection with depth sensors. Colors of the boxes correspond to the type of sensor used: stereo camera as gray, TOF camera as black and Kinect as white. (*) For a clearer visualization, not all algorithms are shown. Salas and Tomasi [29] also employ connected-component analysis, Xia et al. [74] use local chamfer distance and Zhang et al. [81] water filling.

2.2.5 Summary

Regardless of the type of camera, pattern matching is an accurate and popular method to find people in images. Gradient histograms, such as HOG and HOD, have been a major direction for the research. Other recently used features include head-shoulder contour matching, local chamfer distance and histogram of depth difference. Pattern matching has mainly two disadvantages: it is computationally intensive and it does not perform very well in crowded scenes. Both of these challenges have been addressed. First, computational cost has been lowered by intelligently decreasing the number of comparison operations. Second, accuracy in a crowded scene has been improved by estimating crowd density in each pixel, then fusing the results of people detector and density calculator. Another, lighter approach is to first extract foreground pixels, i.e. define which pixels in an image contain humans. After this, groups of multiple people need to be analysed to count and locate individuals.

Taking advantage of highly accurate depth sensors makes foreground extraction an easy task. If the camera is stationary, background subtraction will give most of the foreground pixels with practically no false positives. If the camera is moving, or background scene changing fast, occupancy maps will give a good estimate on the locations of tall objects such as humans.

Group segmentation, on the other hand, is an unsolved problem. In simple cases, where all individuals form their own uniform area in the foreground, connected-component analysis is satisfactory. However, when an area in the foreground consists of multiple persons, more advanced methods are required. Depth images enable much better possibilities to analyse groups, compared to color images. Examples of well-performing group analysis methods include 3D point clustering, depth image segmentation and water filling. It is still unclear which would be the best set of algorithms in different situations, as they have not been compared using a common dataset.

2.3 Multiple target tracking

People detection results in the locations of people in single images. People tracking is the following step after detection that determines who is who in a consecutive set of images. For example, when a person is detected for the first time, he or she is given an ID. The task of tracking is to find which detections in the following images correspond to this same person and label all of these detections with the ID. The result of tracking is a set of trajectories for each person that has been observed. [13]

In its simplest form, tracking is only linking the current detections to the nearest detections on the previous frame. However, this will require that each individual is correctly localized each frame and that the objects never get very close to each other. In practice, people detectors produce false positives, are not always able to find everyone and give inaccurate position estimates for real persons. Therefore, more advanced methods have to be developed.

Perhaps the most challenging situation is when the input data contains the wrong number of people for a compact group. This would be the case if people were so close to each other that they were detected as a single person, or if a person was behind another. This kind of situation is called a merge. When all the individuals are again distinguishable, a so-called split occurs. To maintain correct trajectories for everyone, all the members of the merged group have to be associated with an individual that was observed before the merge (figure 13).

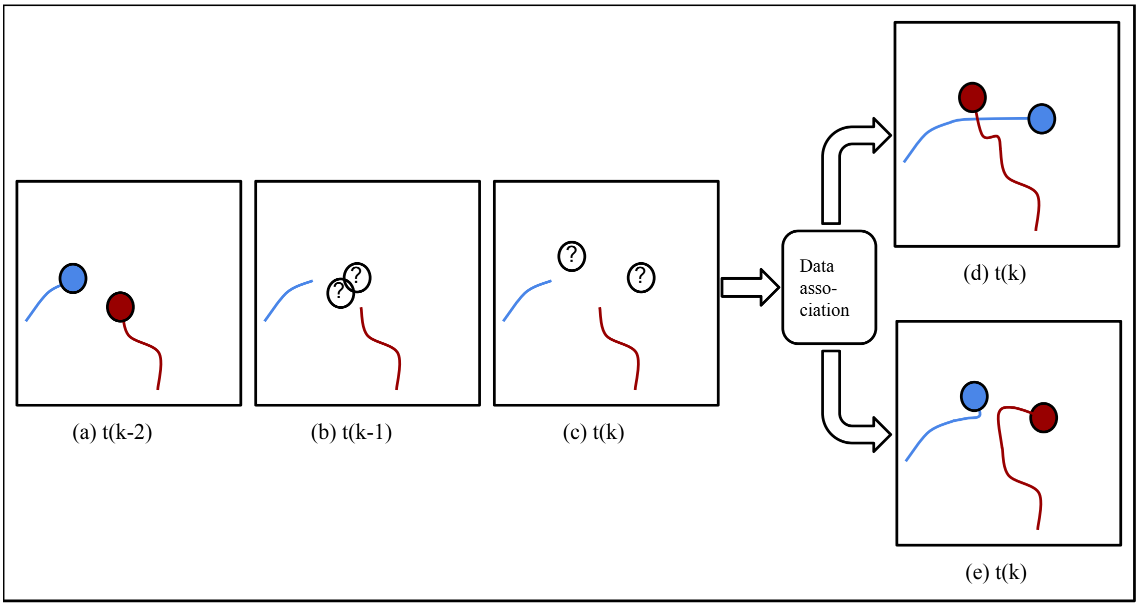


Figure 13: Good data association is important especially in merge-split situations, where tracked objects get very close to each other (b). Data association chooses either option (d) or (e), based on information obtained from (a), (b) and (c).

Data association may use multiple cues to find corresponding detections. As explained in section 2.4, color histograms have been very popular cues. Other possible measures include height, width, shape, speed, texture, and several specialized image features such as SIFT and HOG [13]. It has to be noted that height, shape, width and speed can be measured more accurately using depth information.

According to Pulford [82], the simplest true multiple target tracking algorithm is the nearest-neighbour Kalman filter (NNKF). At each time step, each measurement is assigned to exactly one existing track. Assignment is based on the measurements and a prediction of the track by the Kalman filter. Measurements contain at least the location, but other cues such as velocity and color histogram may be used too. Blackman (2004) states that NNKF works well in the case of widely spaced targets, accurate measurements and few false alarms near the real targets. Seer et al. [78] have shown that a simple nearest-neighbour algorithm is a viable solution for people tracking with Kinect, due to the high quality detection data.

Another simple and effective solution is Joint Probabilistic Data Association (JPDA) by Fortmann et al. [83], where each measurement may contribute to the update of more than one track. In other words, a track is updated by a weighted sum of all nearby observations. However, JPDA has problems with closely spaced targets merging too easily [84]. Also, JPDA assumes a fixed number of targets and cannot initiate or terminate tracks by default [85].

Naturally, assigning measurements to tracks based on only two consecutive time steps will fail at some point, when the input data is not perfect. More robust methods calculate the assignment problem based on multiple time steps. Among the first, Reid [86] has implemented Multiple Hypothesis Tracker (MHT) by a tree of Kalman filters. In case of an uncertain event, MHT will store the most probable assignment combinations for later use, so that the decision can be delayed until more reliable data is available. Various improvements and versions of MHT have been proposed; for further information, see [20].

Although multiple hypothesis trackers have been very effective and widely used, their computational complexity is relatively high since the number of hypotheses grows exponentially over time [87].

Early 2000s, particle filters became popular in multi-target tracking [88]. One of the most successful particle filters is RJMCMC (Reversible Jump Markov Chain Monte Carlo) by Green [89], applied to people tracking by Khan et al. [90], Smith et al. [91] and Choi et al. [92]. Compared to multiple hypothesis tracking, MCMC data association shows remarkable performance under extreme conditions such as handling dense groups or high false alarm rates [87].

The latest trend, however, seems to be global optimization [93, 58, 94, 95, 21], that is, finding an optimal path for each individual, based on the whole sequence of images. Berclaz et al. [21] split videos into batches of 100 frames, resulting in a 4 second delay between image acquisition and tracking results. This kind of delay is often tolerable in surveillance applications. The downside of using a large time frame is computational cost: their algorithm runs barely real-time on a typical desktop PC. Hofmann et al. [96] implemented global data association using a three stage hierarchical track forming strategy. The functionality behind each stage is similar, only parameter settings are different. Tracks are iteratively grown in three steps, which will finally lead to a set of target trajectories (figure 14):

1. Build small tracklets and direct links based on spatial overlap. If only one new detection is very close to an old one, make a link.

2. Correct detection misses and small occlusions by frame skip.

3. Handle long-time occlusions.

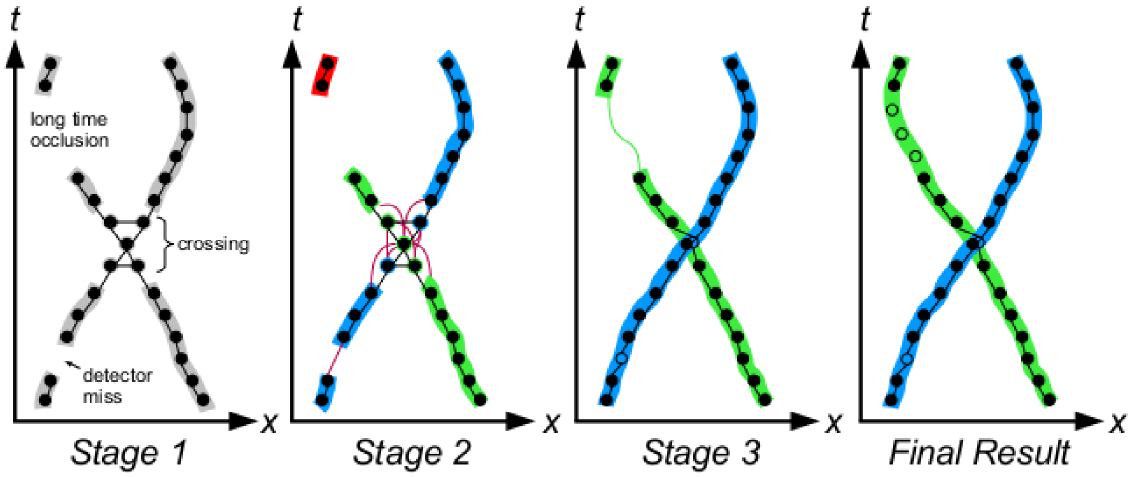


Figure 14: Illustration of hierarchical tracking strateby by Hofmann et al. [96]. Stage 1: build small tracklets and direct links based on spatial overlap. Stage 2: use frame skip to handle small occlusions and misses. Stage 3: handle longtime occlusions. Text and image from [96].

The authors of global optimization trackers have given reasons why their algorithms would be better than particle filter trackers, such as MCMC:

1. *Despite their success, in our experience, those sampling-based methods typically require careful tuning of several meta-parameters, which reduces the generality of systems that rely on it. Besides, they usually look at small time windows because their state space grows exponentially with the number of frame.*

   — Berclaz et al. [21]

2. *Most notable, using a Markov chain in the definition of the transition probabilities would require all information of an object (appearance, motion, etc.) to be present in a single frame. However, both motion and appearance can hardly be captured in just a single frame.*

   — Hofmann et al. [96]

To summarize the current state of people tracking, global optimization has the best results with the highest computational cost, while MCMC particle filters are faster and have a very good tracking accuracy. However, nearest-neighbour tracking is still a viable option if input data from a people detector is reliable, being very fast and easy to implement.

2.4 Data association and re-identification

In order to reliably track an individual over multiple frames and camera views, the same appearance or shape has to be modelled. For example, height and color can be used as cues to determine who is who in different images. Matching persons between frames (intra-camera) is called data association, while finding the same persons in different cameras (inter-camera) is called re-identification. Re-identification is one of the biggest difficulties in building sparse camera networks [97]. Problems include changing illumination, camera poses, occlusions, clothes and apparel. Commonly, it is assumed that people have a constant appearance, i.e. they wear the same clothes and apparel in all cameras. [97]

There are various visual features that can be used to find corresponding persons in different frames. Previously, features have been classified as global and local. Global features describe the whole person, including size, velocity and color. Local features are smaller details of the image. Features are observed with descriptors, such as color histogram, HOG (Histogram of Oriented Gradients), SIFT (Shift-Invariant Feature Transform), GMM (Gaussian Mixture Model) or Haar-wavelet. [97]

State-of-the-art examples of re-identification include Asymmetry-based Histogram Plus Epitome (AHPE) [98], Symmetry-Driven Accumulation of Local Features (SDALF) [99], Custom Pictorial Structures (CPS) [100] and local distance comparison [101]. All of the previous basically rely on body part segmentation (except for local distance comparison) and a hue saturation value color histogram in their own ways. However, body part segmentation is difficult in crowded scenes [81]. Another exception is the work of Bak et al. [102], who have presented an approach called Mean Riemannian Covariance Grid (MRCG) that achieves good performance by observing temporal changes of appearance.

Most of the state-of-the-art algorithms do not address the issue of processing time [103]. Especially CPS is far from useful for real-time applications [101]. While Local Distance Comparison algorithm presented in [101] is much more efficient than CPS and apparently real-time on a desktop PC, it most likely is too slow for low-cost embedded hardware.

A lightweight alternative is to use a simple color-position position histogram [104], or similar approaches proposed by Bird et al. [105] and Albiol et al. [26]. Color position is fast to calculate: the silhouette is first vertically divided into equal parts. Then, the mean color is computed to characterize each part (figure 15). Compared to the classical color histogram, this method leads to better results with the help of spatial information and uses less memory. [104]

Another limitation of many algorithms are camera angle and occlusions. In the previously mentioned papers, performance is mostly evaluated using datasets such as VIPeR [106], ETHZ [107] and CAVIAR4REID [108]. However, these datasets always view people from the side or low angle, and occlusions are quite rare or missing. In real-world use cases it is often necessary to place a people-flow counter as high as possible to avoid occlusions. For people tracking applications, it would be better to evaluate re-identification with more challenging datasets, such as PETS 2009 [109], as Baltieri et al. [110] have done.

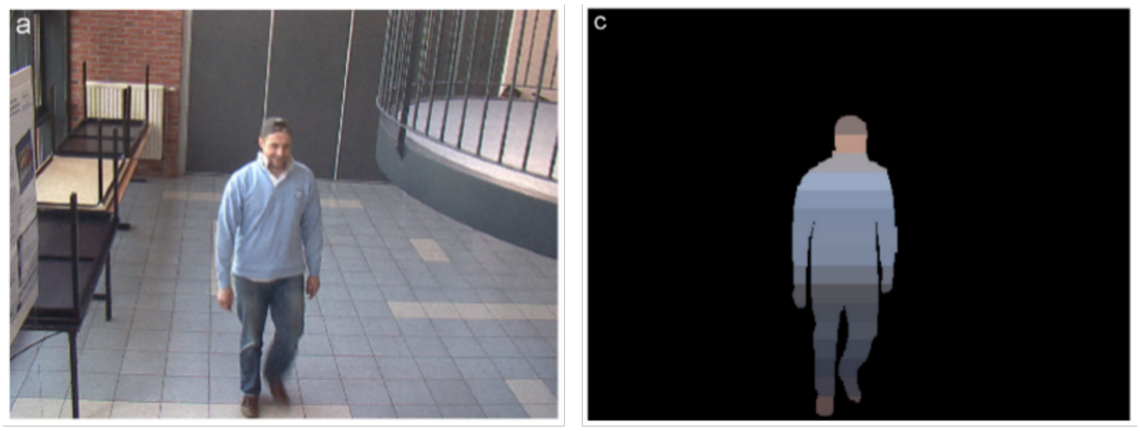


Figure 15: Principle of the color position histogram for re-identification. Left: original, unsegmented image. Right: Colors divided into vertical bins. Image from [104].

Kinect has already been tested in re-identification by Barbosa et al. [111] and Albiol et al. [26]. Barbosa et al. [111] observed ten different shape-based cues, including height estimate, Euclidean distance between torso center and right shoulder, or Geodesic distance between torso center and right hip. They found out that the most informative cues obtained with Kinect were height and torso/legs ratio.

3 Methods

3.1 System requirements

While a big part of the academic research puts a huge effort in maximizing the detection and tracking accuracy, the priorities of this work are elsewhere. Accuracy is of course important, but low development, hardware and installation costs are essential. Hardware and development costs set limitations for algorithm complexity. Installation costs are minimized by automatic calibration: the aim is that each camera node will become fully functional by just plugging in the power chord. In addition to low cost, system design is based on various other requirements.

1. Algorithms need to be robust in different environments, where lighting, surface materials and furniture may vary.

2. Selecting an optimal camera location and angle needs to be easy, while maintaining sufficient accuracy and area coverage.

3. The system has to be widely applicable and extensible to different applications.

4. The system has to be easily scalable. Adding or removing a camera should require very little configuration effort.

The Kinect is an excellent choice for producing high quality data in most lighting conditions, excluding sunlight. However, the data may have significant noise especially as the range increases. Glass and other reflective surfaces are a major cause for failed depth measurements. To achieve robust detection, algorithms that manage with incomplete and erroneous data should be used, or noise needs to be removed by filtering.

Commercial people counting products are often downwards-facing cameras. The reason is clear: placing cameras above the people is an easy way to avoid occlusions. However, as Harville [16] notes, the top view has a limited detection area if the ceiling is low. Therefore, the optimal solution is to enable vertical, horizontal and tilted viewing angles. The selection can be made just before installing a camera.

3.2 System overview

A prototype of a smart camera network for people tracking applications was built for this thesis. A smart camera is a device that not only acquires image data, but produces high-level understanding of the imaged scene [112]. In this case, the smart camera will gain information about detected individuals, including position, height and speed of individuals. The device will then send information to a central server that will parse this data and form information about people flow in a larger scale.

While smart cameras often perform processing on a Digital Signal Processor (DSP) or a Field-Programmable Gate Array (FPGA) [112], it was decided to use an ARM development board. This kind of development board is easier to program and applications are also compatible with PC hardware. For the first development platform, Pandaboard ES [113] was chosen. The Pandaboard is a single-board computer based on the Texas Instruments OMAP4460 processor. The selection is based on a wide software and hardware support, a relatively low cost and a high processing power. Currently, the processing speed of a dual core ARM Cortex-A9 is sufficient for simple image processing algorithms, but it is also easy to upgrade the hardware to next generation embedded processors if needed.

Each Pandaboard is capable of analysing depth data from one camera in real time. The term camera node is used to describe such pair of a camera and a computer, including a power supply and an optional Wi-Fi antenna.

Camera nodes utilise Internet Protocol (IP) to connect to a central server that is typically a PC. However, the server application is lightweight and can be run on one of the camera nodes if needed, even if the node would run people detection software simultaneously. The choice of IP can be justified easily.

1. IP is suitable for both wireless and cable connection.

2. Buildings often already have the hardware needed for an IP network, such as built-in Ethernet cabling and Wi-Fi routers.

3. Possible additional hardware has high availability and low cost.

Demanding image processing tasks are performed on each camera node (figures 16 to 17). Compared to a solution where cameras are connected to a powerful server that processes all images, distributed computing meets the scalability requirement much better. Additionally, data traffic between the cameras and the server is minimal, for no image data needs to be sent.

A similar sensor to Kinect, ASUS Xtion Pro is used, as it is smaller in size and does not require an external power supply. Images are captured and converted to metric depth measures using the OpenNI framework.

In the next sections, algorithms are described in detail. First, simple background subtraction is applied (section 3.4). Then, remaining pixels are converted to a 3D point cloud using the camera calibration (section 3.3). A height map is generated from the point cloud, and its local maxima are considered as human candidates (section 3.5). Finally, the coordinates of the detected people are sent to a server application that tracks individuals over time (section 3.6).

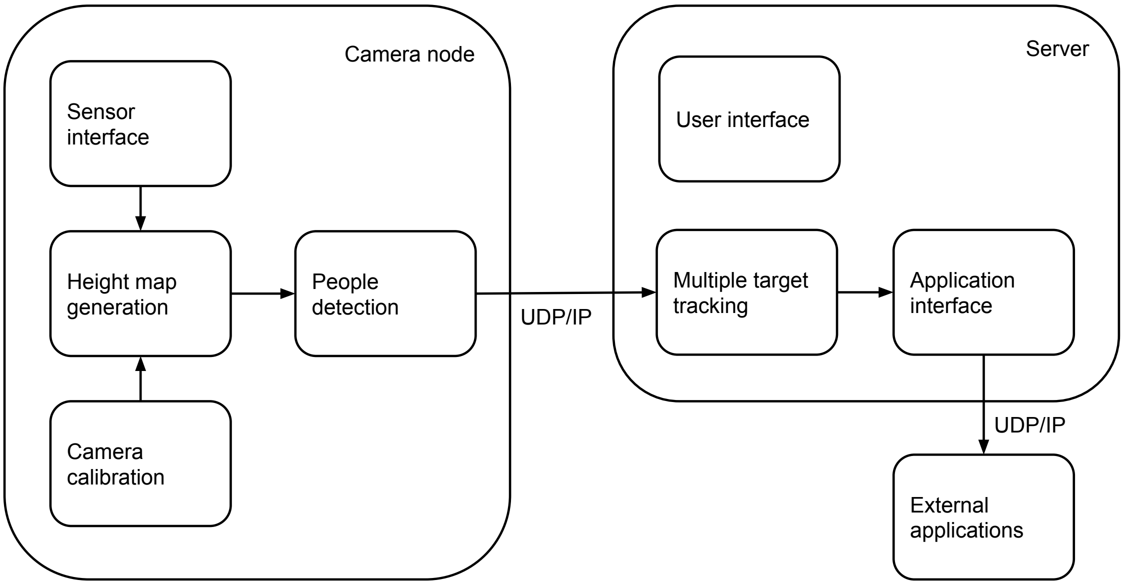


Figure 16: System architecture. Camera node then sends people detections as 3D points over UDP/IP. Server software keeps track of individuals over time. A customizable application interface extracts relevant information of the tracks and sends it to higher level external applications.

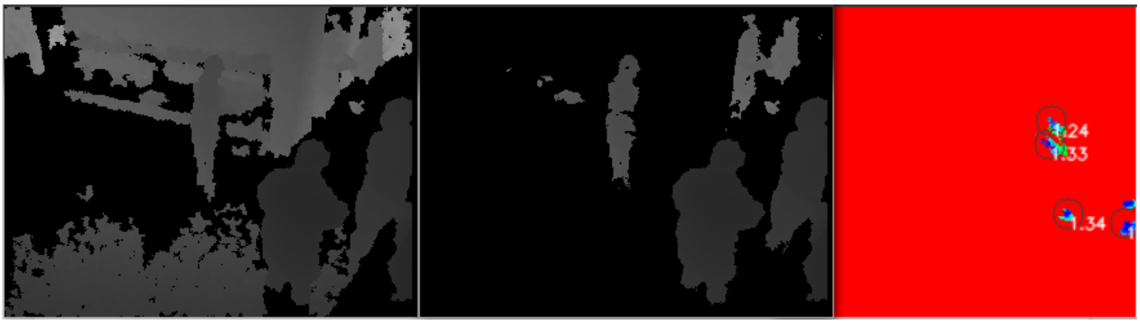


Figure 17: Illustration of the image processing phases. Left: original depth image. Middle: foreground objects. Right: height map, where circles indicate detected humans. Height of the people is written as meters, but it is not correctly measured due to lack of accurate calibration.

3.3 Calibration

The purpose of calibration is to convert measurements into a metric world coordinate system. System calibration consists of three phases.

1. *Camera intrinsic calibration* – conversion from image pixels to metric 3D points.
2. *Camera pose calibration* – transformation of 3D points so that floor corresponds to the surface $z = 0$.
3. *World calibration* – defining the position and angle of each camera in the world coordinate frame, e.g. relative to a floor plan of a building.

For simplicity, an optimal pinhole camera model is used for intrinsic calibration. To obtain a point (x, y, z) from an image pixel (u, v) with a depth value d, equation (1) is used.

$$\begin{cases} x = \frac{(u-c_x)d}{f_x} \\ y = \frac{(v-c_y)d}{f_y} \\ z = d, \end{cases} \tag{1}$$

where (c_x, c_y) are pixel values for the camera optical center and (f_x, f_y) are the focal lengths in each direction, for which values have been obtained from Burrus [114]. Values (equation (2)) are fixed by a factor 2, for a half resolution image from Kinect is used.

$$\begin{cases} f_x = \frac{2}{594.21} \\ f_y = \frac{2}{591.04} \\ c_x = \frac{339.30}{2} \\ c_y = \frac{242.73}{2} \end{cases} \tag{2}$$

For best results, each camera should of course be calibrated separately, but this has not yet been one of the top priority tasks. The selected algorithms are robust to small errors in intrinsic calibration.

Camera pose calibration is needed to transfer the camera to a virtual overhead pose, previously presented in figure 5. Pose calibration has been partly automatised with floor detection. Namely, the camera tilt angle is found by calculating the angle between floor plane and the camera optical axis. Also, taking advantage of depth data enables the camera height (z coordinate) to be measured.

A popular method to find the floor is called Random Sample Consensus (RANSAC). However, RANSAC is computationally intensive, therefore planes are directly found in the depth image using V-disparity by Labayrade et al. [115]. The implemented algorithm is similar to the original work, with the following custom additions. As detecting the floor is handled by searching surfaces, false positives occur if other surfaces, such as walls and tables, cover a major part of the depth image. A score for each plane is calculated based on multiple attributes, after which the plane with the biggest score is selected as the floor plane. These attributes are:

1. Height of the lowest part of the plane in the image.
2. Amount of image pixels that form the plane.
3. Extra score is given if the plane normal is within manually set boundaries corresponding to the tilt angle of the camera. These may be changed depending on the knowledge on how the cameras will be installed.

Finally, the last phase of calibration is to transfer coordinates from camera coordinate frame to world coordinate frame. Distances between camera nodes could be learned automatically [116]. However, if the camera install locations are known, automatic calibration is not required. With limited time, each camera is manually given a position and a pan angle in world coordinate frame.

3.4 Pre-processing

As noted by Harville [16], foreground extraction from depth images is a simple task. Taking advantage of this, only a minimal, yet efficient background subtraction technique is implemented. Background image bg is constructed by taking N (typically $20 - 200$) consecutive depth frames and keeping the minimum non-zero depth value for each pixel.

Background subtraction is then performed by comparing further depth images to bg pixel by pixel. A pixel with depth value d is marked as foreground, if

$$bg(j,i) + threshold < d, \tag{3}$$

where $threshold$ is set to 20 cm.

It is clear that this kind of simple approach will fail if furniture is moved in the scene, as the background model is not automatically updated. The use of different background models, such as GMM or ViBe, have been considered to overcome this problem, but they are not implemented at this point.

3.5 People detection

The most accurate people detection methods of the present day are often based on gradient histograms. When using Kinect, one of the most popular approaches is to combine Histogram of Oriented Gradients with Histogram of Oriented Depths. There are many problems in gradient histograms (discussed in section 2.2.2), but perhaps the most severe problem is computational complexity. The goal in this work is to make low-cost people detecting devices, therefore lighter algorithms are needed.

Hordern and Kirchner [67] note that processing 2D projections of 3D data is computationally efficient. Examples include plan-view projections [16] (figure 5) and silhouette projections [80, 67] combined with head-shoulder template matching. Plan-view projections such as occupancy map or height map have the advantages of view-invariance and computational efficiency. Silhouette matching, on the other hand, produces less false positives for other moving objects that are as tall as humans.

Plan-view projections seem very attractive, as they meet the requirements for arbitrary camera angles and processing time. Preliminary tests show a good performance for both height and occupancy maps. A purely height map based approach is selected for the first evaluation, because it is usually more robust to partial occlusions in e.g. crowded scenes [16]. The best results could be reached by combining these two plan-view statistics [16], but this kind of an approach is left for further implementation.

For detecting people in a height map the most obvious method is chosen. As head is usually the tallest part of a human, local maxima are found from the height image. The height map is a 200 x 200 pixel top-view projection of the point cloud. Slight post-processing is needed to reduce false positives. First, all maxima outside of the range 0.8 – 2.1 meters are omitted. Then, circumference of the underlying object is calculated and all small objects are removed. The minimum contour length depends on the scale of the scene, typically being 10 – 35 pixels.

The previously described method is fast, simple and powerful, but there are at least three flaws. First, using only height map discards information by grouping multiple points in the vertical direction as one height value. This may lead to faraway, occluded targets to be removed as false positives. Second, looking only at local maxima will produce misses or false positives in dense crowds. For example, if there is an adult next to a child, the child's head will be lower than the adult's shoulder, and the child will be left unnoticed. Third, other large moving objects may be falsely detected as humans.

3.6 People tracking

Seer et al. [78] have shown that even a simple nearest-neighbour algorithm is able to maintain tracks for long periods of time, because it is easy to detect people accurately from Kinect depth data. Nearest-neighbour is also fast to implement and fast to compute, therefore a reasonable choice for this prototype system.

The tracking algorithm is an implementation of the nearest-neighbour matching. However, the tracker is not yet completed; the current implementation lacks many important features. First, a constant frame rate is assumed. A part of the parameters, such as the time needed for deleting an obsolete track, are measured in numbers of frames instead of actual seconds. Second, merging and splitting groups is not reliably implemented, resulting in identity switches in crowded situations. Therefore it was decided not to describe the tracking algorithm in more detail. Still, the tracking performance will be evaluated briefly using the same data sets

3.7 Evaluation

The emphasis of the thesis is to evaluate how well the presented approach works in a generic indoor environment. Most effort was put in optimizing and evaluating the people detection algorithm, but tracking performance was measured as well. Testing is driven by the following requirements and limitations:

1. Cameras should be installable in arbitrary poses.
2. People must be detected even in dense crowds, where partial occlusions are common.
3. No manual calibration is allowed.
4. The scene may not be illuminated by direct sunlight, not even through windows.
5. A small portion of floor needs to be visible.

Four data sets from an office building were gathered (table 5), keeping the above requirements in mind. See appendix A for illustration of the recording scenes. The recordings were carried out during one week with the Pandaboard devices located in three different positions. All cameras were aligned in different heights and angles. Recording was automatically triggered after detecting a certain amount of people in the scene ($1-5$, depending on the place), then recording a certain amount of time ($15-60$ seconds, depending on the place). During the recording the number of people was not counted, as it would have decreased the frame rate. Checking if there are still enough people in the scene before resuming the recording causes the image sequences to be slightly discontinuous. Afterwards, many of the sequences with only one person were removed to provide more challenging data. Also, some of the images with an empty scene were removed. The number of people in each dataset is presented in figure 18.

Manually annotating ground truth, i.e. real locations of people, to thousands of images is time-consuming. An annotation tool was developed, which tries to make marking ground truth as easy as possible (figure 19). People are marked on the depth images by holding the mouse cursor over the person's head, while browsing through the images. The tool will automatically paint all points that are closer than 30 cm from the point under the cursor (3D coordinate comparison) and within a user-configurable 2D circle.

The ground truth is exported to images, where blue pixels indicate the ground truth. Different blue value is used for every individual person, so that the first person gets the RGB color value $(0, 0, 255)$, second gets the color $(0, 0, 254)$ and so on. This way, as many as 255 individuals can be explicitly placed in a single image by changing only the blue value. The ground truth is saved as an image where certain areas are painted as ground truth, to allow manual fixing with an image manipulation program. Also, for someone who wants to take advantage of the dataset, it is better to have ground truth information that is not dependent on the calibration, as 3D points would be.

Table 3: Gathered image sequences for evaluating the presented people detection algorithm.

Name	**Description**	**Frames**
Hallways	Intersection of two hallways. Small amount of people mainly walking.	4556
Cafeteria	Cafeteria tables where people are mainly sitting and eating. Occasionally people leave the tables or walk past the cafeteria.	10050
Buffet #1	People taking food from a buffet table, a small number of people queuing for the cashier (cashier not visible in the scene).	3300
Buffet #2	Similar to buffet sequence #1, but with more people.	3750

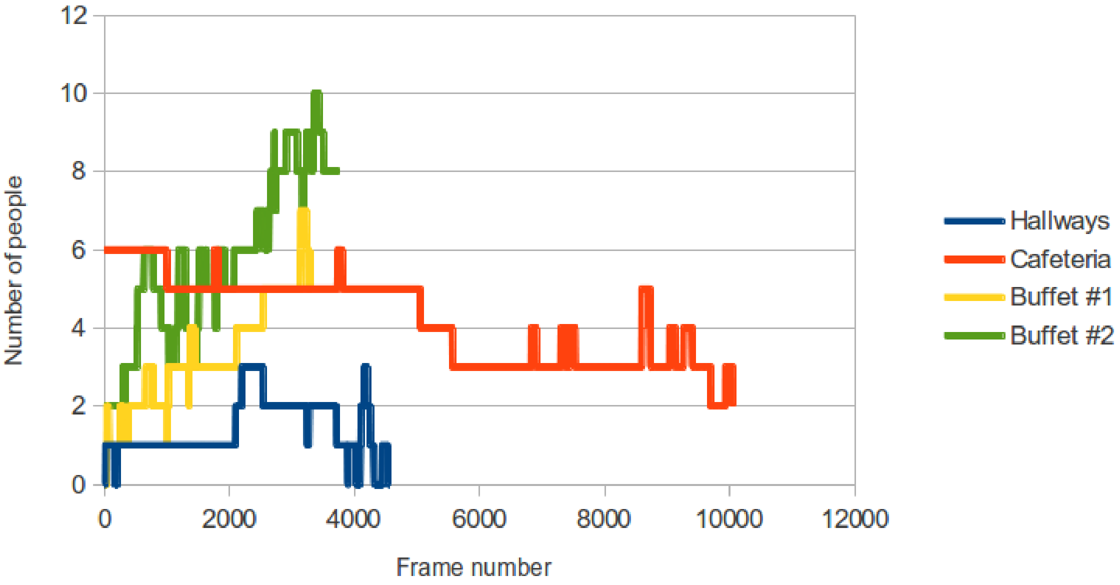


Figure 18: Number of people in each dataset.

Finally, the ground truth images are compared with the people detector output (figure 20). The output is saved as 3D coordinates, which are back-projected on the original depth image using the same calibration. A detection is considered to be successful if there is a ground truth pixel within 4 pixels of the detection pixel. This kind of tolerance was allowed to overcome rounding errors, changed coordinates caused by noise filtering, and annotation errors, such as the one in figure 21.

Based on the comparison of detections and ground truth, precision and recall rates are calculated from the amount of true positives TP, false positives FP and

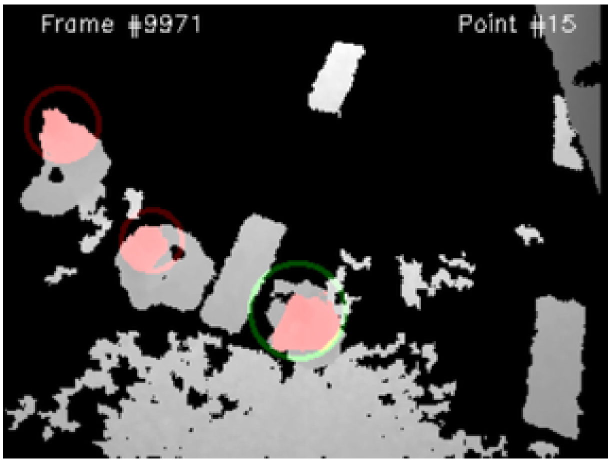


Figure 19: Illustration of the annotation tool. User moves the circles with the mouse to mark the correct locations of humans. Pixels marked with red are part of the ground truth. The aim is to mark only the upper part of the body, e.g. head and shoulders.

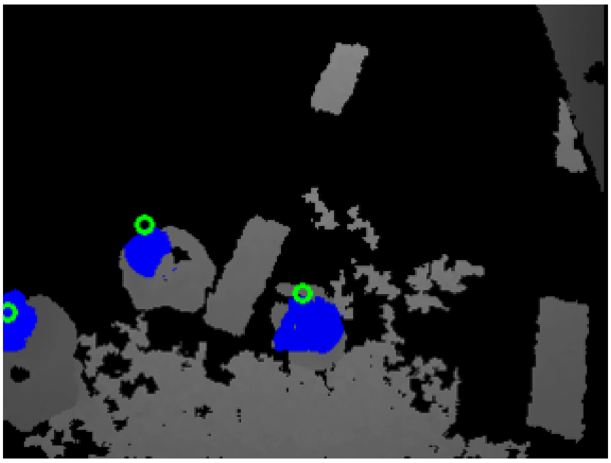


Figure 20: An evaluation tool that compares output from people detector (green circles) with ground truth images (people marked blue). Image from 'Cafeteria' dataset.

false negatives FN (equations (4) and (5)).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

TP is the number of cases, where exactly one detection corresponds to a ground truth person. FN is the number of misses, i.e. cases where a real person was undetected. FP is the sum of false and duplicate detections. Note that multiple occurrences of these events may be counted on one frame.

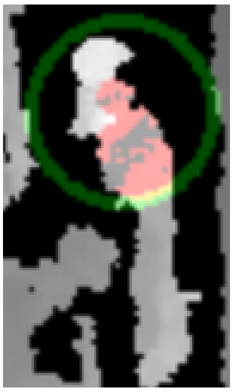


Figure 21: An example of a failed annotation. The whole upper part of the human body should be uniformly marked with red. Instead, annotated area is scattered due to high noise at a long distance.

The people tracking algorithm was shortly evaluated with the same datasets (table 5), but without the help of an automatic tool. Instead, the results were manually written down while observing a video that was slowed down to a 25 % speed. Evaluation metrics described by Li et al. [117], Hofmann et al. [96] were used. These are Identity Switches (IDS), Track Fragments (FM), Mostly Tracked (MT), Partly Tracked (PT) and Mostly Lost (ML) (see table 4).

Table 4: Evaluation metrics for people tracking. Table from [117].

Name	**Definition**
GT	Number of ground truth trajectories.
MT	Mostly tracked: Percentage of GT trajectories which are covered by tracker output for more than 80 % in length.
ML	Mostly lost: Percentage of GT trajectories which are covered by tracker output for less than 20 % in length. *The smaller the better.*
PT	Partially tracked: 1.0 − $MT - ML$.
FM	Fragments: The total of No. of times that a ground truth trajectory is interrupted in tracking result. *The smaller the better.*
IDS	ID switches: The total of No. of times that a tracked trajectory changes its matched GT identity. *The smaller the better.*

4 Results

4.1 Automatic calibration

Performance evaluation of the automatic calibration was not originally planned, but it was tested while recording the data sets for the people detector. For this reason, there are not enough samples to verify the functionality, yet a rough estimate on the accuracy level can be given. In all of the three datasets gathered, automatic floor detection was successful. Unmodified automatic pose calibration was also used when evaluating the people detection algorithm. The general impression is that floor detection works most of the time, but not nearly always. For example, while testing the automatic recording software situations occurred where other similarly aligned planes, such as tables or cupboards, were marked as the floor plane. Another challenge are reflective surfaces that are viewed on from a narrow angle: most of the time depth data is lost or it is very noisy. Five successful and one failed attempt are shown in figure 22.

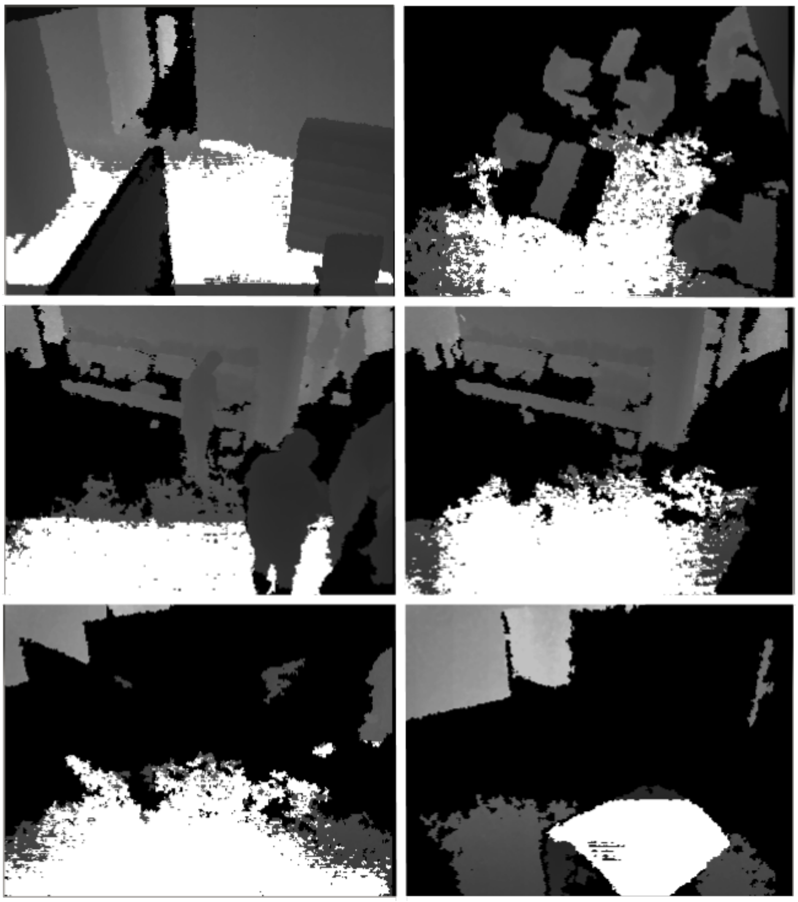


Figure 22: Results of the automatic floor detection, detected floor pixels as white. On the top and middle rows are presented the scenes of the evaluated datasets. On the bottom row are one successful and one failed sample from other tests. See appendix A for color images of the scenes.

4.2 People detection

The presented algorithm performs well in the three arbitrarily selected scenes of an office building. In general, precision and recall were over 90 % with reasonable parameters. However, the 'Hallways' set has a poor performance due to a cleaning trolley that is in the scene most of the time. Other causes for false positives were trays and bags carried by people, or trays set on the table. Even if the background subtraction algorithm is supposed to ignore objects closer than 20 cm to the background, the tables in dataset 'Cafeteria' were largely invisible to the depth camera. As there was no background in these spots, even thin trays were falsely considered as foreground objects. Such problem could arise even from setting a piece of paper on a reflective surface.

False negatives, on the other hand, were mainly caused by crowded situations, overused pre-processing or occlusions. If the distance between two persons' heads is smaller than the local maxima search window size, the shorter one will not be detected. Partial occlusions caused by other persons were not a major problem, but people standing on the border of the image with only less than half of the body visible, were not detected very well. Additionally, annotation and evaluation tools may have caused approximately 2 % error to both precision and recall. This is because annotation was semi-automatic and evaluation fully automatic, so the results on each frame were not manually checked.

The required balance between recall and precision may vary depending on the application; in some cases false positives are more tolerable than in others. To serve all applications, the algorithm is evaluated with 4 common sets of parameters, namely the local maxima search window size and minimum blob contour size (table 5). Small values for these parameters will give a good true positive rate but more false positives, and vice versa. The result for each dataset is a precision-recall curve (figure 23).

4.3 People tracking

The simple nearest-neighbour tracking shows mediocre results, but no extensive conclusions can be drawn from this test: datasets contained paths of only about 65 individuals. As can be seen in table 6, people could be mostly tracked through their paths in the image. However, there was a notably high amount of discontinuous tracks (i.e. fragments) and identity switches. These were mainly caused by three reasons. First, people walking fast or changing the direction quickly caused the detections to be too far away from the tracked trajectory. As a result, the track was broken into two pieces (fragments) and the identity of the person was not maintained. Second, the people detection algorithm produces unreliable results in long distances due to depth data noise. This causes both missed detections and position errors in the detection results. Third, the lack of cues for data association (e.g. height, size, direction, color) often caused an identity switch of two neighbouring persons.

Table 5: Parameters and results of the people detection algorithm.

Dataset	**Local maxima search window size [mm]**	**Minimum blob contour [mm]**	**Precision**	**Recall**
Hallways	200	400	0.232	0.876
	300	600	0.334	0.880
	400	800	0.484	0.860
	500	1000	0.494	0.613
Cafeteria	200	400	0.712	0.980
	300	600	0.920	0.909
	400	800	0.944	0.737
	500	1000	0.968	0.601
Buffet #1	200	400	0.771	0.992
	300	600	0.904	0.978
	400	800	0.968	0.912
	500	1000	0.974	0.858
Buffet #2	200	400	0.825	0.918
	300	600	0.907	0.895
	400	800	0.931	0.825
	500	1000	0.951	0.727

Table 6: Tracking results. See table 4 for acronym explanations.

Dataset	**GT**	**MT**	**PT**	**ML**	**FM**	**IDS**
Hallways	8	87.5 %	12.5 %	0.00 %	3	3
Cafeteria	21	100 %	0.00 %	0.00 %	4	4
Buffet #1	11	100 %	0.00 %	0.00 %	8	8
Buffet #2	25	76.0 %	20.0 %	4.00 %	16	16

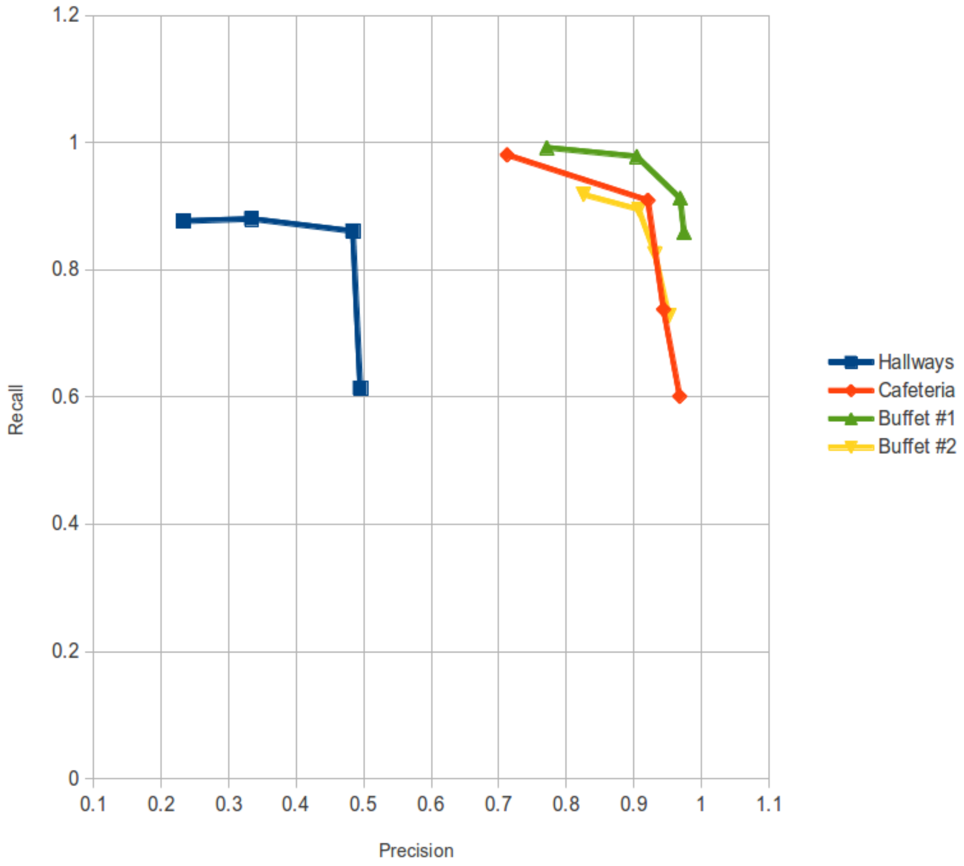


Figure 23: Results of the people detection. Three datasets reach 90 % precision and recall simultaneously, but the 'Hallways' set has a tremendous amount of false positives from a cleaning trolley.

4.4 Computational efficiency

People can be detected from a 320 x 240 pixel depth image approximately 20 to 30 times per second on the OMAP4460, while reserving less than 50 megabytes memory. The frame rate varies vastly depending on the number of foreground objects, so that an empty scene can be processed more than 30 frames per second and if all pixels belong to the foreground, frame rate is only 20. The processing times for different algorithm steps are presented in table 7.

Nodes cause only a small network load. The data rate for transferring the locations of 16 people or less is 15 kilobytes per second. Also, data packets are not sent if there are no people in the scene.

Table 7: Processing time for the suggested people detection algorithm, measured while processing the image sequence 'Buffet #1'. Duration of image processing per frame depends on the number of foreground pixels. It has to be noted that OpenNI sensor driver runs as a different process, which consumes a significant amount of processing power that is not presented in this table.

	Time [ms], average	**Time [ms], maximum**
Extract foreground	4	6
Generate height map	20	32
Detect people	5	8
Others	1	2
Total	**30**	**48**

5 Summary

This thesis consists of mainly two parts. First, an extensive survey of image-based people detection was presented, with an emphasis on detection from depth images. A total of 21 publications using depth-based methods were reviewed. Three common categories of algorithms were identified and discussed: gradient histograms, silhouette matching and plan-view projections. As a conclusion, there does not seem to be a clear winner. Most of the researchers combine multiple different methods, often resulting in a very good accuracy.

Second, a system for detecting people indoors with a network of depth cameras is presented and evaluated. The experiments show that this approach results in a good accuracy. Also, the system is easy to install: calibration is mostly automatic and cameras can be positioned arbitrarily. The detection algorithm, based on a height map, is significantly faster compared to pattern matching and most other methods presented in the literature. The hardware requirements for simple algorithms are low, decreasing costs and making the approach more appealing.

Currently, there are many limitations, but most of them can be overcome. First, objects illuminated by sunlight are invisible to the Kinect. A solution is to take advantage of the color camera and detect moving objects by background subtraction. Second, the background model should be updated automatically to reduce false positives from large, movable objects. Another option is to classify objects by motion features. Third, the tracking system should be improved by replacing the simple nearest-neighbour algorithm with either MCMC or global optimization of multiple frames. Finally, a better picture of individual's trajectories within a building could be obtained with re-identification. A lightweight solution, such as the color position histogram by Cong et al. [104], would be suitable for an embedded system.

Depth information has proven to have value in people detection. It has made extracting foreground objects very simple, provided additional cues for pattern matching and enabled new kind of algorithms. Counting and locating people in a dense group remains a challenging task. Multiple depth-based methods for group segmentation have been presented, including water filling, clustering or convex hull segmentation. However, it remains unknown which is the optimal solution in terms of accuracy and computational complexity, as they have not been compared with the same data sets.

References

[1] ShopperTrak. *ShopperTrak Solutions.* 2013. URL: `www.shoppertrak.com/products` (visited on 11/14/2013).

[2] Biodata Ltd. *Use CCTV to Count People.* 2013. URL: `http://videoturnstile.com` (visited on 11/14/2013).

[3] Alex Leykin and Mihran Tuceryan. “Tracking and activity analysis in retail environments”. In: (2005).

[4] M. Valera and SA Velastin. “Intelligent distributed surveillance systems: a review”. In: *Vision, Image and Signal Processing, IEE Proceedings-.* Vol. 152. IET, 2005. Chap. 2, pp. 192–204.

[5] Axiomatic Technology Ltd. *Our customers.* 2013. URL: `http://www.peoplecounting.co.uk/our-customers.html` (visited on 11/14/2013).

[6] DILAX Intelcom GmbH. *Applications of passenger counting systems for public transit.* 2012. URL: `http://www.dilax.net/people-counting-systems-passenger-counting-systems-visitor-count/passenger-counting-systems-for-public-transportation-people-count-in-trains/examples-for-passenger-counting-systems-in-public-transportations.html` (visited on 11/14/2013).

[7] Zhengyou Zhang. “Microsoft kinect sensor and its effect”. In: *Multimedia, IEEE* 19.2 (2012), pp. 4–10.

[8] John Krumm et al. “Multi-camera multi-person tracking for easyliving”. In: *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on.* IEEE, 2000, pp. 3–10.

[9] Luis M. Fuentes and Sergio A. Velastin. “People tracking in surveillance applications”. In: *Image and Vision Computing* 24.11 (2006), pp. 1165–1171.

[10] Sofia Zaidenberg et al. “Group interaction and group tracking for video-surveillance in underground railway stations”. In: *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011).* 2011.

[11] Dirk Helbing and Pratik Mukerji. “Crowd disasters as systemic failures: analysis of the Love Parade disaster”. In: *EPJ Data Science* 1.1 (2012), pp. 1–40.

[12] Minghao Wang and Xiaolin Hu. “Data assimilation in agent based simulation of smart environment”. In: *Proceedings of the 2013 ACM SIGSIM conference on Principles of advanced discrete simulation.* ACM, 2013, pp. 379–384.

[13] Thiago Teixeira, Gershon Dublon, and Andreas Savvides. “A survey of human-sensing: Methods for detecting presence, count, location, track, and identity”. In: *ACM Computing Surveys* 5 (2010).

[14] Brickstream Corporation. *Brickstream 3D Data Sheet.* 2013. URL: `http://www.brickstream.com/products/brickstream-devices/` (visited on 12/13/2013).

[15] Infrared Integrated Systems Ltd. *People Counting Products.* 2013. URL: http://www.irisys.co.uk/people-counting/products/ (visited on 12/13/2013).

[16] Michael Harville. "Stereo person tracking with adaptive plan-view statistical templates". In: *Proc. ECCV Workshop on Statistical Methods in Video Processing.* 2002, pp. 67–72.

[17] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE, 2005, pp. 886–893.

[18] Piotr Dollár et al. "Pedestrian detection: An evaluation of the state of the art". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.4 (2012), pp. 743–761.

[19] Olivier Barnich and Marc Van Droogenbroeck. "ViBe: A universal background subtraction algorithm for video sequences". In: *Image Processing, IEEE Transactions on* 20.6 (2011), pp. 1709–1724.

[20] Samuel S. Blackman. "Multiple hypothesis tracking for multiple target tracking". In: *Aerospace and Electronic Systems Magazine, IEEE* 19.1 (2004), pp. 5–18.

[21] Jerome Berclaz et al. "Multiple object tracking using k-shortest paths optimization". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.9 (2011), pp. 1806–1819.

[22] Microsoft. *Kinect.* 2011. URL: http://www.xbox.com/en-us/kinect/ (visited on 12/20/2013).

[23] Robert B Fisher, Kurt Konolige, and Artificial Intelligence Center. "Handbook of Robotics Chapter 22-Range Sensors". In: 2008.

[24] John Iselin Woodfill et al. "The tyzx deepsea g2 vision system, ataskable, embedded stereo camera". In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on.* IEEE. 2006, pp. 126–126.

[25] e-con Systems Pvt. Limited. *Capella - Stereo Vision Camera Reference Design.* 2013. URL: http://www.e-consystems.com/Stereo-Vision-Camera.asp (visited on 12/21/2013).

[26] A Albiol, J Oliver, and JM Mossi. "Who is who at different cameras: people re-identification using depth cameras". In: *IET computer vision* 6.5 (2012), pp. 378–387.

[27] Jungong Han et al. "Enhanced Computer Vision with Microsoft Kinect Sensor: A Review". In: *IEEE Transactions on Cybernetics* (2013).

[28] Oliver Sidla et al. "Pedestrian detection and tracking for counting applications in crowded situations". In: *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on.* IEEE, 2006, pp. 70–70.

[29] Joaquín Salas and Carlo Tomasi. "People detection using color and depth images". In: Pattern Recognition. Springer, 2011, pp. 127–135.

[30] Luciano Spinello and Kai O. Arras. "People detection in RGB-D data". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on.* IEEE, 2011, pp. 3838–3843.

[31] Thiago Teixeira and Andreas Savvides. "Lightweight people counting and localizing in indoor spaces using camera sensor nodes". In: *Distributed Smart Cameras, 2007. ICDSC'07. First ACM/IEEE International Conference on.* IEEE, 2007, pp. 36–43.

[32] Chris Stauffer and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking". In: *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* Vol. 2. IEEE, 1999.

[33] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. "A survey of advances in vision-based human motion capture and analysis". In: *Computer Vision and Image Understanding* 104.2 (2006), pp. 90–126.

[34] Marc Van Droogenbroeck and Olivier Paquot. "Background subtraction: experiments and improvements for vibe". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on.* IEEE, 2012, pp. 32–37.

[35] Liang Zhao and Charles E. Thorpe. "Stereo-and neural network-based pedestrian detection". In: *Intelligent Transportation Systems, IEEE Transactions on* 1.3 (2000), pp. 148–154.

[36] Kourosh Khoshelham and Sander Oude Elberink. "Accuracy and resolution of kinect depth data for indoor mapping applications". In: *Sensors* 12.2 (2012), pp. 1437–1454.

[37] Piotr Dollár, Serge Belongie, and Pietro Perona. "The Fastest Pedestrian Detector in the West." In: *BMVC.* Vol. 2. 2010. Chap. 3, p. 7.

[38] Greg Mori, Serge Belongie, and Jitendra Malik. "Efficient shape matching using shape contexts". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.11 (2005), pp. 1832–1837.

[39] Edgar Seemann et al. "An Evaluation of Local Shape-Based Features for Pedestrian Detection." In: *BMVC.* Vol. 5. Citeseer, 2005, p. 10.

[40] Christian Wojek and Bernt Schiele. "A performance evaluation of single and multi-feature people detection". In: Pattern Recognition. Springer, 2008, pp. 82–91.

[41] Harry G. Barrow et al. In: *Parametric correspondence and chamfer matching: Two new techniques for image matching* (1977).

[42] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. "A general framework for object detection". In: *Computer Vision, 1998. Sixth International Conference on.* IEEE, 1998, pp. 555–562.

[43] Paul Viola and Michael Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.* Vol. 1. IEEE, 2001, I–511–I–518 vol. 1.

[44] David G Lowe. “Object recognition from local scale-invariant features”. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on.* Vol. 2. Ieee. 1999, pp. 1150–1157.

[45] David G. Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[46] Bo Wu and Ram Nevatia. “Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors”. In: *International Journal of Computer Vision* 75.2 (2007), pp. 247–266.

[47] Payam Sabzmeydani and Greg Mori. “Detecting pedestrians by learning shapelet features”. In: *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on.* IEEE, 2007, pp. 1–8.

[48] Navneet Dalal, Bill Triggs, and Cordelia Schmid. “Human detection using oriented histograms of flow and appearance”. In: *Computer Vision–ECCV 2006.* Springer, 2006, pp. 428–441.

[49] William Robson Schwartz et al. “Human detection using partial least squares analysis”. In: *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE. 2009, pp. 24–31.

[50] Qiang Zhu et al. “Fast human detection using a cascade of histograms of oriented gradients”. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on.* Vol. 2. IEEE, 2006, pp. 1491–1498.

[51] Navneet Dalal. *INRIA Person Dataset.* 2005. URL: `http://pascal.inrialpes.fr/data/human/` (visited on 12/21/2013).

[52] Rodrigo Benenson et al. “Pedestrian detection at 100 frames per second”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 2903–2910.

[53] Xiaoyu Wang, Tony X. Han, and Shuicheng Yan. “An HOG-LBP human detector with partial occlusion handling”. In: *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 32–39.

[54] Stefan Walk et al. “New features and insights for pedestrian detection”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 2010, pp. 1030–1037.

[55] Ching-Yao Chan, Fanping Bu, and Steven Shladover. *Experimental vehicle platform for pedestrian detection.* California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2006.

[56] Duc Fehr et al. “Counting people in groups”. In: *Advanced Video and Signal Based Surveillance, 2009. AVSS’09. Sixth IEEE International Conference on.* IEEE, 2009, pp. 152–157.

[57] Prahlad Kilambi et al. “Estimating pedestrian counts in groups”. In: *Computer Vision and Image Understanding* 110.1 (2008), pp. 43–59.

[58] Li Zhang, Yuan Li, and Ramakant Nevatia. “Global data association for multi-object tracking using network flows”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[59] Tao Zhao and Ramakant Nevatia. “Bayesian human segmentation in crowded situations”. In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2003, pp. II–459.

[60] Mikel Rodriguez et al. “Density-aware person detection and tracking in crowds”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2423–2430.

[61] Pedro F Felzenszwalb et al. “Object detection with discriminatively trained part-based models”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.9 (2010), pp. 1627–1645.

[62] Takeo Kanade et al. “A stereo machine for video-rate dense depth mapping and its new applications”. In: *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR’96, 1996 IEEE Computer Society Conference on*. IEEE. 1996, pp. 196–202.

[63] Christopher Eveland, Kurt Konolige, and Robert C. Bolles. “Background modeling for segmentation of video-rate stereo sequences”. In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*. IEEE, 1998, pp. 266–271.

[64] Trevor Darrell et al. “Integrated person tracking using stereo, color, and pattern detection”. In: *International Journal of Computer Vision* 37.2 (2000), pp. 175–185.

[65] David Beymer. “Person counting using stereo”. In: *Human Motion, 2000. Proceedings. Workshop on*. IEEE. 2000, pp. 127–133.

[66] Sergiu Nedevschi, Silviu Bota, and Corneliu Tomiuc. “Stereo-based pedestrian detection for collision-avoidance applications”. In: *Intelligent Transportation Systems, IEEE Transactions on* 10.3 (2009), pp. 380–391.

[67] Daniel Hordern and Nathan Kirchner. “Robust and efficient people detection with 3-d range data using shape matching”. In: (2010).

[68] Ruijiang Luo and Yan Guo. “Real-time stereo tracking of multiple moving heads”. In: *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 2001. Proceedings. IEEE ICCV Workshop on*. IEEE. 2001, pp. 55–60.

[69] Junji Satake and Jun Miura. “Stereo-based tracking of multiple overlapping persons”. In: *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2581–2585.

[70] Sho Ikemura and Hironobu Fujiyoshi. “Real-time human detection using relational depth similarity features”. In: *Computer Vision–ACCV 2010*. Springer, 2011, pp. 25–38.

[71] Shengyin Wu, Shiqi Yu, and Wensheng Chen. “An attempt to pedestrian detection in depth images”. In: *Intelligent Visual Surveillance (IVS), 2011 Third Chinese Conference on*. IEEE, 2011, pp. 97–100.

[72] Qing Tian et al. “Human Detection using HOG Features of Head and Shoulder Based on Depth Map.” In: *Journal of Software (1796217X)* 8.9 (2013).

[73] Max Bajracharya et al. “A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle”. In: *The International Journal of Robotics Research* 28.11-12 (2009), pp. 1466–1485.

[74] Lu Xia, Chia-Chih Chen, and JK Aggarwal. “Human detection using depth information by Kinect”. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*. IEEE, 2011, pp. 15–22.

[75] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. “Tracking people within groups with rgb-d data”. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2101–2107.

[76] Jose-Juan Hernandez-Lopez et al. “Detecting objects using color and depth segmentation with Kinect sensor”. In: *Procedia Technology* 3 (2012), pp. 196–204.

[77] Ching-Tang Hsieh et al. “A Kinect-based people-flow counting system”. In: *Intelligent Signal Processing and Communications Systems (ISPACS), 2012 International Symposium on*. IEEE, 2012, pp. 146–150.

[78] Stefan Seer, Norbert Brandle, and Carlo Ratti. “Kinects and Human Kinetics: A New Approach for Studying Crowd Behavior”. In: *arXiv preprint arXiv:1210.2838* (2012).

[79] Richard O. Duda, Peter E. Hart, and David G. Stork. “Pattern classification”. In: *New York: John Wiley, Section* 10 (2001), p. l.

[80] Huiyuan Fu, Huadong Ma, and Hongtian Xiao. “Real-time accurate crowd counting based on RGB-D information”. In: *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 2685–2688.

[81] Xucong Zhang et al. “Water Filling: Unsupervised People Counting via Vertical Kinect Sensor”. In: *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*. IEEE, 2012, pp. 215–220.

[82] GW Pulford. “Taxonomy of multiple target tracking methods”. In: *Radar, Sonar and Navigation, IEE Proceedings-*. Vol. 152. IET, 2005. Chap. 5, pp. 291–304.

[83] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. “Multi-target tracking using joint probabilistic data association”. In: *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*. Vol. 19. IEEE, 1980, pp. 807–812.

[84] RJ Fitzgerald. “Track biases and coalescence with probabilistic data association”. In: *Aerospace and Electronic Systems, IEEE Transactions on* 6 (1985), pp. 822–825.

[85] Songhwai Oh et al. “A fully automated distributed multiple-target tracking and identity management algorithm”. In: *Proc. AIAA Guidance, Navigation, and Control Conference*. 2005.

[86] Donald Reid. “An algorithm for tracking multiple targets”. In: *Automatic Control, IEEE Transactions on* 24.6 (1979), pp. 843–854.

[87] Songhwai Oh, Stuart Russell, and Shankar Sastry. “Markov chain Monte Carlo data association for general multiple-target tracking problems”. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*. Vol. 1. IEEE, 2004, pp. 735–742.

[88] Tao Zhao and Ramakant Nevatia. “Tracking multiple humans in complex situations”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.9 (2004), pp. 1208–1221.

[89] Peter J. Green. “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination”. In: *Biometrika* 82.4 (1995), pp. 711–732.

[90] Zia Khan, Tucker Balch, and Frank Dellaert. “MCMC-based particle filtering for tracking a variable number of interacting targets”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.11 (2005), pp. 1805–1819.

[91] Kevin Smith, Daniel Gatica-Perez, and J-M Odobez. “Using particles to track varying numbers of interacting people”. In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005, pp. 962–969.

[92] Wongun Choi, Caroline Pantofaru, and Silvio Savarese. “Detecting and tracking people using an rgb-d camera via multiple detector fusion”. In: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1076–1083.

[93] Jerome Berclaz, Francois Fleuret, and Pascal Fua. “Robust people tracking with global trajectory optimization”. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE, 2006, pp. 744–750.

[94] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. “Globally-optimal greedy algorithms for tracking a variable number of objects”. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 1201–1208.

[95] Joao F. Henriques, Rui Caseiro, and Jorge Batista. "Globally optimal solution to multi-object tracking with merged measurements". In: *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011, pp. 2470–2477.

[96] Martin Hofmann, Michael Haag, and Gerhard Rigoll. "Unified Hierarchical Multi-Object Tracking using Global Data Association". In: PETS, 2013.

[97] Mingli Song, Dachent Tao, and Stephen J. Maybank. "Sparse Camera Network for Visual Surveillance–A Comprehensive Survey". In: *arXiv preprint arXiv:1302.0446* (2013).

[98] Loris Bazzani et al. "Multiple-shot person re-identification by chromatic and epitomic analyses". In: *Pattern Recognition Letters* 33.7 (2012), pp. 898–903.

[99] Michela Farenzena et al. "Person re-identification by symmetry-driven accumulation of local features". In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* IEEE, 2010, pp. 2360–2367.

[100] Dong Seon Cheng et al. "Custom Pictorial Structures for Re-identification." In: *BMVC*. Vol. 2. 2011. Chap. 5, p. 6.

[101] Guanwen Zhang et al. "Local distance comparison for multiple-shot people re-identification". In: Computer Vision–ACCV 2012. Springer, 2013, pp. 677–690.

[102] Slawomir Bak et al. "Multiple-shot human re-identification by mean riemannian covariance grid". In: *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on.* IEEE, 2011, pp. 179–184.

[103] Riccardo Satta, Giorgio Fumera, and Fabio Roli. "Fast person re-identification based on dissimilarity representations". In: *Pattern Recognition Letters* 33.14 (2012), pp. 1838–1848.

[104] D-N Truong Cong et al. "People re-identification by spectral classification of silhouettes". In: *Signal Processing* 90.8 (2010), pp. 2362–2374.

[105] Nathaniel D Bird et al. "Detection of loitering individuals in public transportation areas". In: *Intelligent Transportation Systems, IEEE Transactions on* 6.2 (2005), pp. 167–177.

[106] UCSC Computer Vision Lab. *VIPeR: Viewpoint Invariant Pedestrian Recognition.* 2007. URL: `http://vision.soe.ucsc.edu/node/178` (visited on 12/16/2013).

[107] W.R. Schwartz and L.S. Davis. *ETHZ Dataset for Appearance-Based Modeling.* 2009. URL: `http://homepages.dcc.ufmg.br/~william/datasets.html` (visited on 12/16/2013).

[108] Loris Bazzani. *CAVIAR4REID.* 2011. URL: `http://www.lorisbazzani.info/code-datasets/caviar4reid/` (visited on 12/16/2013).

[109] James Ferryman. *PETS 2009 Benchmark Data.* 2009. URL: `http://www.cvg.rdg.ac.uk/PETS2009/a.html` (visited on 12/16/2013).

[110] Davide Baltieri, Roberto Vezzani, and Rita Cucchiara. “SARC3D: a new 3D body model for people tracking and re-identification”. In: *Image Analysis and Processing–ICIAP 2011*. Springer, 2011, pp. 197–206.

[111] Igor Barros Barbosa et al. “Re-identification with RGB-D Sensors”. In: *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 433–442.

[112] D. Shi and Serge Lichman. “Smart Cameras, A Review”. In: *Proceedings of.* 2005, pp. 95–100.

[113] pandaboard.org. *Pandaboard ES*. 2013. URL: `http://pandaboard.org/content/pandaboard-es` (visited on 01/20/2014).

[114] Nicolas Burrus. *Kinect Calibration*. 2011. URL: `http://burrus.name/index.php/Research/KinectCalibration` (visited on 12/20/2013).

[115] Raphael Labayrade, Didier Aubert, and J-P Tarel. “Real time obstacle detection in stereovision on non flat road geometry through”. In: *Intelligent Vehicle Symposium, 2002. IEEE*. Vol. 2. IEEE. 2002, pp. 646–651.

[116] Paulo Freitas, Paulo Menezes, and Jorge Dias. “Online Topological Mapping of a Sparse Camera Network”. In: Technological Innovation for Value Creation. Springer, 2012, pp. 229–240.

[117] Yuan Li, Chang Huang, and Ram Nevatia. “Learning to associate: Hybrid-boosted multi-target tracker for crowded scene”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2953–2960.

A Dataset scenes

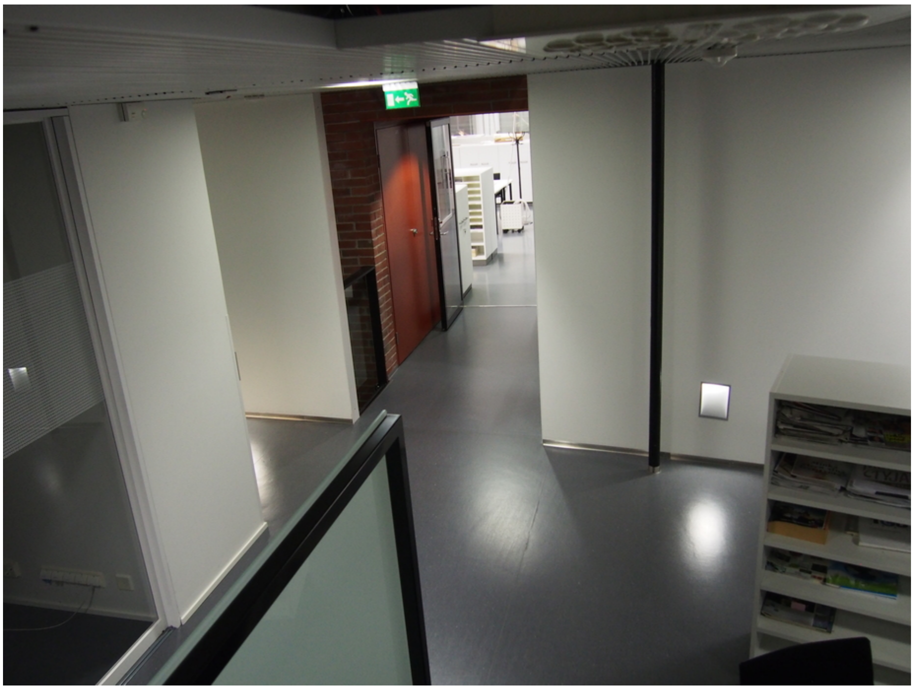


Figure A1: Scene of the dataset 'Hallways'.

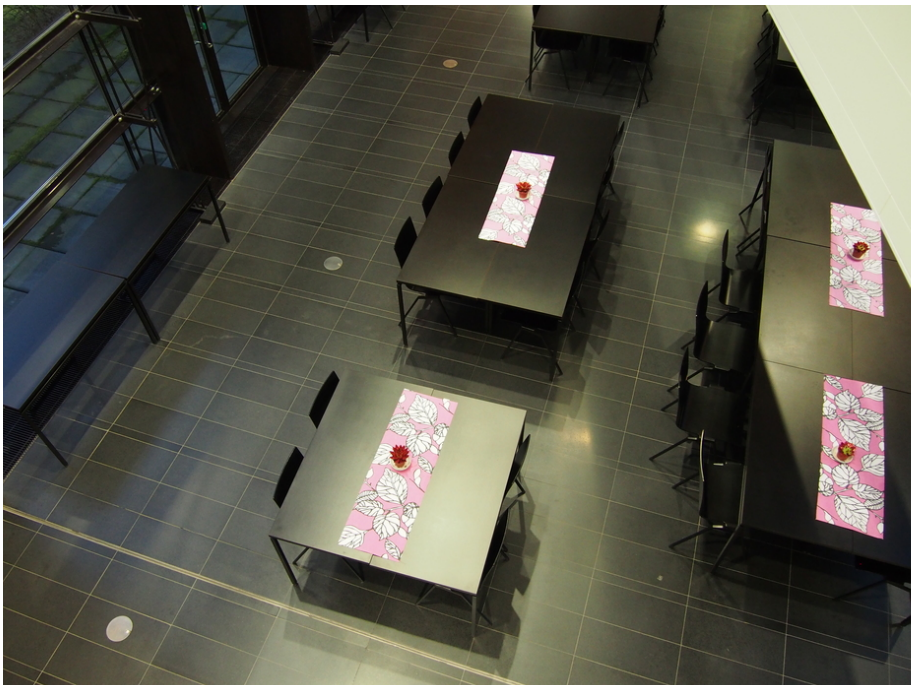


Figure A2: Scene of the dataset 'Cafeteria'.

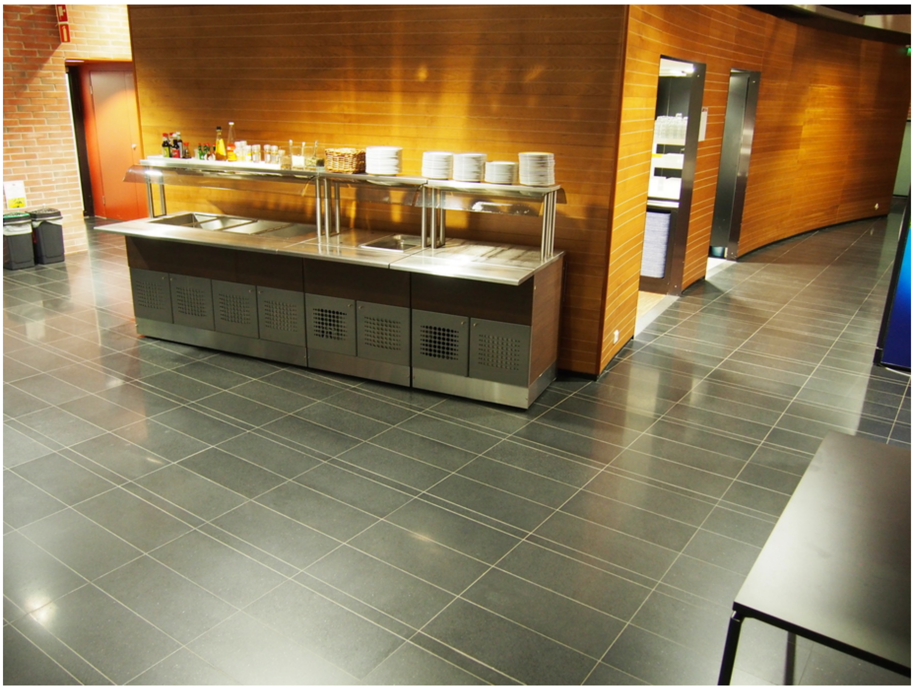


Figure A3: Scene of the datasets 'Buffet #1' and 'Buffet #2'.

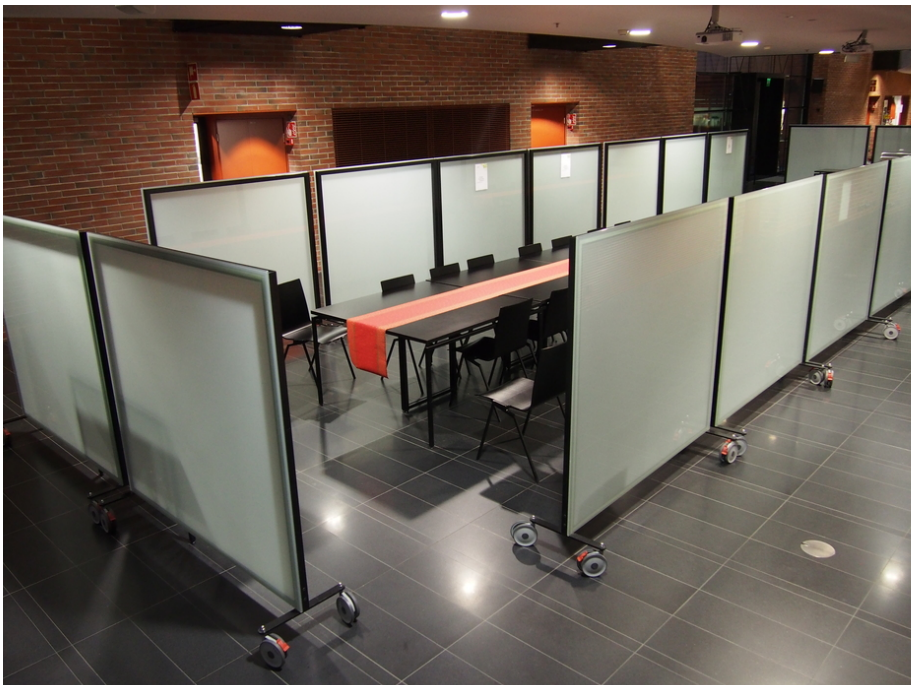


Figure A4: Scene of a floor detection test.

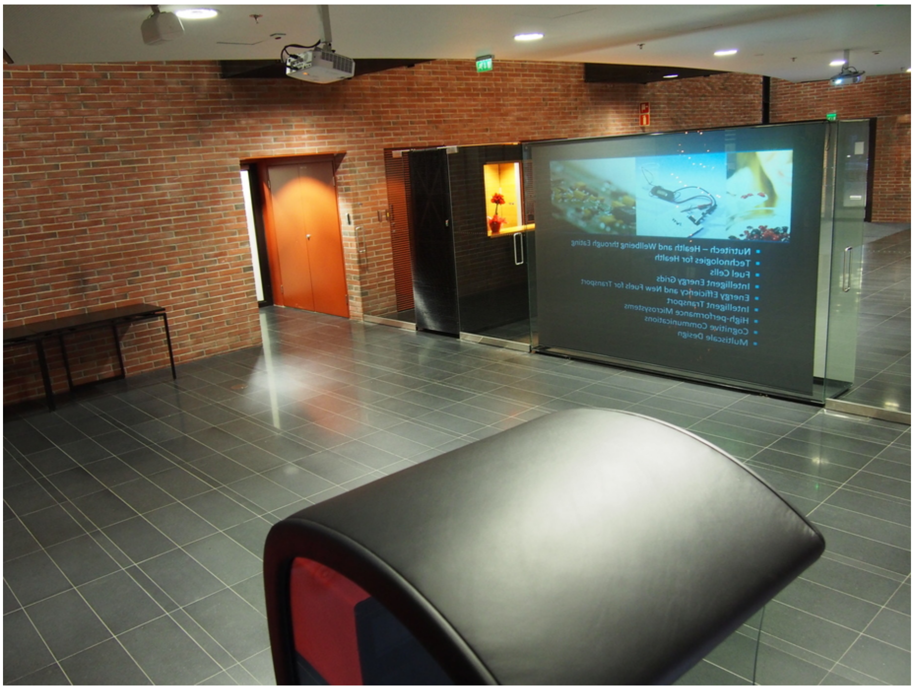


Figure A5: Scene of a floor detection test. Automatic floor detection failed due to the chair on the foreground.