Università degli Studi di Salerno

Université de Lorraine

FACULTY OF ENGINEERING

Master Thesis in Computer Engineering

People Counting and Height Estimation from Overhead Depth Images

---

Supervisors:

Prof. Mario Vento

Prof. Salvatore Antoine Tabbone

Co-supervisor:

Dr. Antonio Greco

Candidate:

Antonio Terrone

Matr.

0622700283

Academic year 2015/2016

Ai miei cari.

Summary

Index of Figures

Index of Tables

1 Introduction

1.1 Problem Formulation

Counting automatically the number of people passing a specific point is a function of paramount importance in applications such as surveillance, monitoring, and interaction between humans and machines. For instance, imagine a civil protection situation taking place in a building: people continuously enter and leave when suddenly an alarm sounds, indicating that the building must be evacuated. One can imagine how useful it is for people in charge of the evacuation procedure to query an automated monitoring system to figure out how many people are still in the building and in what areas.

Other possible examples are very realistic: estimating the crowd density in public places can help managers identify unsafe situations and regulate traffic appropriately; the public museum can control the number of people entering according to the real-time people flow information.

Another possible application is in the transport sector. It could be necessary to count and estimate how many people get on the bus in one bus stop for example. In the past, however, passenger count was mostly done manually, was both labor, and cost intensive. So, it is important to develop an automatic method for counting passengers.

In addition, the retail domain represents one of the most typical scenarios where such systems are used. Indeed, information regarding the number of person that are

in a shop or in a mall is very relevant for several purposes: it allows to better allocate the staff during the day and the week and the peak hours, to correlate the affluence of customers with the marketing campaigns or with the selling performance of similar shops in the same area, and so on.

The problem, unfortunately, is not so easy to fix. In fact, in the examples described it is possible occur in various difficulties. One approach to identify a person using a computer is based on analysis of captured images from a camera. Many methods need a face image for the identification process. However, face images are not always captured correctly from a camera.

For example, in a traffic-choked situation, other persons might occlude the face of a target person.

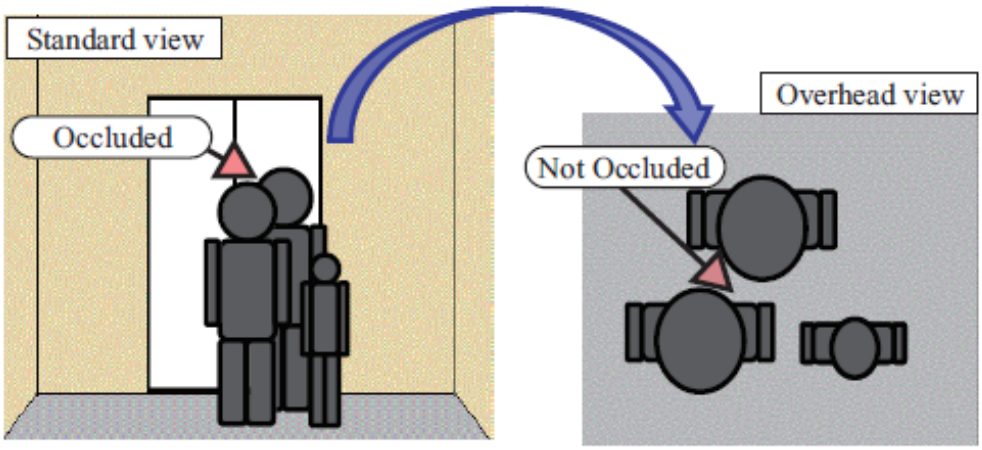


Figure 1 – Occlusion problem

In the Figure 1 above, it is possible to see the occlusion problem and one possible solution that is to adopt an overhead camera. In this work, in fact, we focused on this.

With this solution it will also has other advantages: privacy issue is reduced because the camera does not capture the face image. Furthermore, the restriction of the location of a camera is reduced because the camera does not need to capture the person's face.

On the other hand, information from overhead camera is not always enough. In fact, if the final goal of the system is the identification of a person, the lack of information for the identification method leads to the decrease of the accuracy.

For the goal of this thesis, the face of the people is not important. So we use vertical camera since occlusion problem is automatically solved and in addition we can use the information that the Microsoft Kinect give with the depth video to analyze depth images for people counting and to estimate their height.

A depth camera provides a gray level image where the intensity of the pixel is linearly related to the distance from the camera of that part of the framed object which the pixel belongs to (the higher is the distance from the camera the higher is the intensity value of the pixel).

The proposed method is based on the detection of the heads that allows obtaining these two results. This idea is based on the fact that with the zenithal depth camera the head is always closer to the sensor than other parts of the body and so it is possible use it for these goals. Detecting people's head equals to finding the suitable local minimum regions in the depth image.

The vertical depth information generated by Kinect sensor can simplify the people-counting problem, but there are still problems in real application. This is because people in the same scene may have various scales or depth information, and the crowded people will make a complex depth map with multiple local extremum. Besides, the raw 3D data from Kinect sensor have many noises, which makes the depth map to be discontinuous. So it is difficult to achieve good performance with the traditional clustering method such as mean shift.

Figure 2 – Depth Image from overhead camera

1.2 Sensor used

Kinect sensor is a horizontal bar connected to a small base with a motorized pivot. The device features an RGB camera, depth sensor and multi-array microphone running proprietary software, which provide full-body 3D motion capture, facial recognition and voice recognition capabilities (this not always).

The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor, which captures video data in 3D under any ambient light conditions.

Reverse engineering has determined that the Kinect's various sensors output video at a frame rate of ~9 Hz to 30 Hz depending on resolution. The default RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280x1024 (at a lower frame rate) and other color formats such as UYVY. The monochrome depth sensing video stream is in VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity. The Kinect can also stream the view from its IR camera directly (i.e.: before it has been converted into a depth map) as 640x480 video, or 1280x1024 at a lower frame rate. The area required to play Kinect is roughly 6 m^2, although the sensor can maintain tracking through an extended range of approximately 0.7–6 m (2.3–19.7 ft.). The sensor has an angular field of view of 57° horizontally and 43° vertically, while the motorized pivot is capable of tilting the sensor up to 27° either up or down. The horizontal field of the Kinect sensor at the minimum viewing distance of ~0.8 m (2.6 ft) is therefore ~87 cm (34 in), and the vertical field is ~63 cm (25 in), resulting in a resolution of just over 1.3 mm (0.051 in) per pixel.

2 State-of-the-art

Many methods present in literature have in the people counting the final goal. There are various principal ways to approach to this work of detection and counting:

- using the traditional RGB sensor;
- using the depth sensor;
- using thermal sensor;
- using stereovision sensor;

The first two classes are the ones that concern us more than the other two. The last two ways, or better, the last two sensors have some significant disadvantages that affect the final performance.

The thermal sensors are very expensive and often they have the characteristic that they work as a long-range sensor: this is a problem because it is inconvenient to apply practically and often it does not makes no sense. Furthermore, in many cases it is not possible to specify this range.

On the other hand, also the stereovision sensors have the problem that they cost too much. They have also another serious problem: the performance in the real-time approach are not comparable with those of other sensors.

The results of the numerous works that use the first two sensors are interesting on all the points of view.

As regards the use of RGB cameras, Satarupa Mukherjee et al. (1) propose a novel framework for counting passengers in a railway station. They work with video cameras mounted on the ceilings of the entrance and exit hallways of different stations; they detect every person using Hough circle when he or she enters the field of view. The person is then tracked using optical flow until (s)he leaves the field of view. The framework has three components: people detection, tracking and validation.

A noteworthy novelty in this framework is the introduction of self-validation technique after completing the detection and tracking algorithm. Most of the techniques available in the literature skip the validation step. Here, we emphasize that validation technique cannot be ignored since all the automatic object detection techniques developed till date produce a significant number of false alarms. Hough or HOG based object detection technique generates two types of false alarms: clutter detected as people and duplicates, which are detecting different body parts of the same person. To eradicate these two issues they propose an approximate median (AM) based background subtraction method and measure the ratio of overlap of two trackers in the spatio-temporal domain for rejecting clutter and duplicate trackers respectively and they name the proposed validation framework as spatio-temporal validation (STV).

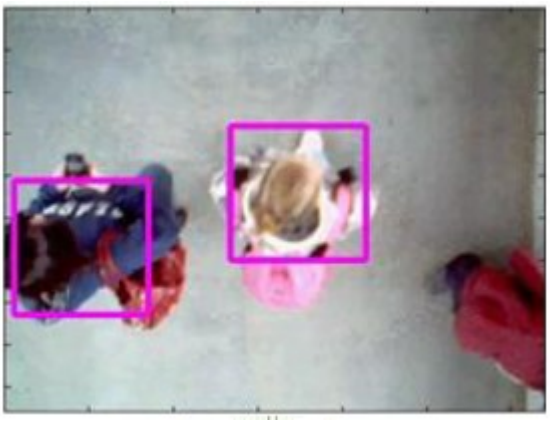


Figure 3 – RGB image from overhead camera

Thou-Ho (Chao-Ho) Chen, Tsong-Yi Chen and Zhi-Xian Chen (2) in their work, addresses the problem of determining the number of pedestrians passing bi-directionally through a gate (or door). They think that the moving direction of the pedestrian can be recognized by tracking each people-pattern with an analysis of its HSI histogram. To improve the accuracy of counting, the color vector extracted from the quantized histograms of intensity or hue is introduced to refine the early counting.

Chao-Ho Chenc et at. (3) propose cost-effective people counter for a crowd of moving people by using a zenithal video camera. To obtain a more accurate people count, the two-stage segmentation is developed for extracting each person from a crowd. Firstly, a crowd is segmented by frame-difference technique, followed by morphological processing and region growing. Then, a connected-component labeling method is used to generate many individual people-patterns from the segmented crowd. People-image features, such as the area, height, and width of each people-pattern, are analyzed in order to correctly segment each person from each individual people-pattern. Finally, each person segmented is tracked until touching the base-line and then is counted.

Bozzoli, Cinque and Sangineto (4) use a single commercial low-cost camera mounted on the ceiling of a public station close to the controlled gate. First, they use a pair of consecutive frames to compute the image optical flow. Motion detection based on gradient images is used for reliable foreground detection insensitive to lighting changes. Connected edge segments are then used to exclude human artifacts and filter out optical flow errors. The corrected optical flow projected onto the line segments is used to estimate the number of people passing through a set of target lines.

Barandiaran, Murguia and Boto (5) using a single overhead mounted camera, with their system count the number of people going in and out of an observed area. Counting is performed by analyzing an image zone composed by a set of virtual counting lines. The system uses multiple independent counting lines, it is not based on the order in which the lines were crossed to know if the person is going in or out, which can be problematic in some circumstances, instead the optical flow is used. Another important improvement of this system against other methods in literature, like (3) for example, is that it does not use morphological operators, because trying to separate a group of people into individuals is not always possible.

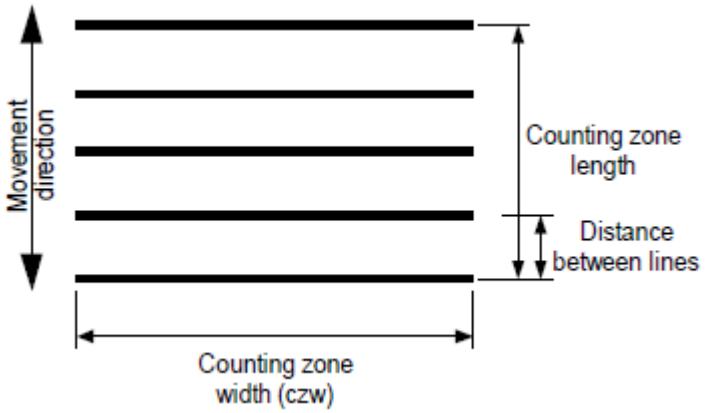


Figure 4 – The counting zone with lines

The system is applicable to indoor scenarios, like corridors or entrances where people can go in or out always in the same direction, i.e. in a parallel way to the expected movement direction. The algorithm is divided in three different steps. Firstly, motion is detected, and moving regions extracted. Then, counting is accomplished by each line. Finally, a global analysis of the results obtained for each line is performed.

Antic´ et al. (6) presented an efficient and reliable approach to automatic people segmentation, tracking and counting, designed for a system with an overhead mounted (zenithal) camera. Upon the initial block-wise background subtraction, k-means clustering is used to enable the segmentation of single persons in the scene. The number of people in the scene is estimated as the maximal number of clusters with acceptable inter-cluster separation. Tracking of segmented people is addressed as a problem of dynamic cluster assignment between two consecutive frames and it is solved in a greedy fashion. Thei experimental results suggest that the proposed method is able to achieve very good results in terms of counting accuracy and execution speed (real-time).

Senem Velipasalar et al. (7) also proposed a system that uses only one camera that is mounted overhead. The *person-size bounds* defining the interval for the size of a single person are the inputs to the system, which are learned in an automatic way. The graphical interface lets the user define a ROI in the camera view so that only the people who pass through the ROI are counted. Two-level hierarchical tracking is employed. For cases not involving merges or splits, a fast blob tracking method is used. In order to deal with interactions among people in a more thorough and reliable way, the system uses the mean shift tracking algorithm. Using the first-level blob tracker in general, and employing the mean shift tracking only in the case of merges and splits saves power and makes the system computationally efficient.

Rossi and Bozzoli (8) with their system want to count the number of people crossing a counting line using a fixed video camera. A motion detection module first determines whether any person has entered the scene; a tracking module combining prediction and matching then follows people until they reach the counting line. Two important constraints were set in order to facilitate the functioning of the system: (1) the camera is mounted vertically with respect to the floor plan; (2) objects enter the scene along only two directions (top and bottom side of the image). The system detects and labels regions where significant motion is observed. Regions are then tracked using a matching operation based on a minimization distance criteria. Finally, an interpretation step based on clustering of regions is performed to determine how many people correspond to the regions being tracked.

Silva de Almeida and Cunha de Melo (9) in their work introduced two methods for people counting. The first one is divided into people segmentation, tracking and counting, developed for a system using a zhenital camera. The initial step consists of block-wise background subtraction, followed by k-means clustering to allow segmentation of single persons in the scene. The number of people in the scene is estimated as the maximal number of clusters with acceptable inter-cluster separation. Tracking of segmented people is addressed as a problem of dynamic cluster assignment between two consecutive frames and it is solved in a greedy fashion. The second it is a method to count people based on analysis of multiple lines. The first part of this algorithm is to detect the movement of people, and regions through which they pass are extracted. From this, the count is performed by virtual lines. Finally, it examines the results for each line.

Serrano-Cuerda et al. (10) proposed a system built from INT3-Horus, a multi-agent based framework for intelligent monitoring and activity interpretation. The system uses an indoor overhead video camera that detects people moving freely in a hall or room. The people counting system is flexible in detecting individuals as well as groups. Counting is independent of the trajectories and possible occlusions of the humans present in the scene.

INT3-Horus is conceived as a framework to carry out monitoring and activity interpretation tasks. This is an ambitious goal given the huge variety of scenarios and activities that can be faced. The framework establishes a set of operation levels where clearly defined input/output interfaces are defined. Inside each level, a developer places his/her code, encapsulated in a module in accordance with the operation performed. The purpose of this framework is easy and fast code integration and generation of real-world systems selecting the best combination to solve a problem from the available modules.

Chunhui Tang et al. (11): the pedestrian detection with a vertical view camera is taken into account, which is usually used to count the pedestrian number. The target characteristic is quite different from traditional one though their targets seem the same. The pedestrian head in vertical view is remarkable and a histogram of oriented gradients (HOG) descriptor is introduced for the head detection. The proposed descriptor originates from traditional pedestrian detection and it is reformed and adaptive in their task. This descriptor is firstly applied to detect pedestrian heads in vertical view.

In the work of Jung-Ming Wang, Sei-Wang Chen et al. (12), fisheye cameras are mounted on the top and capture image from top to down. Foreground people figures are then extracted using the simulation of their human vision system: since human vision system is robust to the illumination changing, their system can be applied in outdoor environments. In the people counting part, human regions are tracked and counted based on a graph matching algorithm. Tracking results are used to determine the direction of region movement based on unary and binary features.

Always Jung-Ming Wang but with Li-Kai Lee et al. (13) proposed a very similar system of that in (12). For people detection, an image segmentation method based on k-means clustering is employed to extract human figures. In order to use in different of illumination conditions, they use region merging to remove shadows of each object. With this approach, their system can be applied in outdoor environments. In the people counting part, human regions are tracked and counted based on a graph matching algorithm and also the results of tracking are used at the same way and for the same purpose.

Antonio Albiol et al. (14) presented an approach to count the number of people that enters or leaves metro trains using zenith cameras located on the station ceilings. This is a challenging scenario where usually people crowd around the train doors, and therefore it is not possible a direct approach that segments and counts individual. The proposed technique is based on a statistical analysis of the flow obtained from the motion vectors at corner points. The features that they use are corner points and their associated motion vectors. Corner points are those with high

spatial gradient and high curvature. They use a modification of the Harris algorithm. They restrict the corner extraction to the area of the platform that is closer to the train. To estimate the motion vector for each corner with respect to its previous frame, they have used standard block-matching with a block-size of 7 pixels and the sum of absolute differences as the similarity measure. Notice that the corners are not tracked. In fact, the exact corner points may change from frame to frame. All what it is assumed is that a person has an average number of corners. They define the flow as the vertical component of the motion vectors. The flow can be instantaneous or accumulated within a time interval. In general, it will take several frames for a person to cross the counting area.

Kulrapat Jaijing, Pakorn Kaewtrakulpong and Supakorn Siddhichai (15) also used an overhead camera. With their work, they propose Snake algorithm with effective external energy function and a half-circle template initialization for modeling passengers. First, they use background subtraction, then weight map was calculated by applying the Gaussian filter to the edge information extracted from the foreground image. Snake algorithm is an edge-based model widely used for object shape modeling and tracking. It does not require any training process. A general Snake is a curve, which moves through the spatial domain of an image to minimize an energy function. The performance of Snake algorithm depends heavily on the initialization. They present the use of a half-circle template, which could effectively initialize the Snake close to the edge boundary of head and shoulder area. The template was placed in the area of interest, which was defined as 40% of rectangular area cropped on the moving objects of the extracted binary foreground image. Nevertheless, in some cases, the half circle template was inappropriately initialized to two adjacent moving objects instead of one leading to the incorrect Snake fitting.

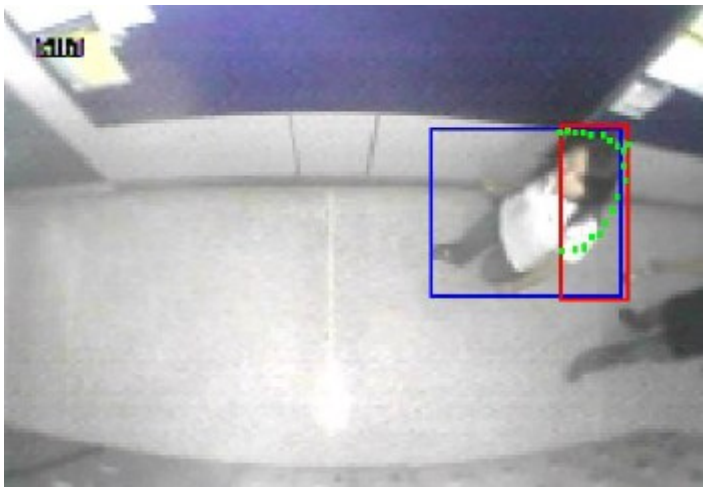


Figure 5 – Positive example of template in use

Gardel Vicente et al. (16) shows the algorithm implementation for a field-programmable gate array (FPGA)-based design for people counting using a low-level head-detection method. To achieve all these requirements, the use of specific hardware (HW) components is proposed. The main advantage of this type of systems when compared with an embedded image processor is the high frame rate achieved. Video applications on an embedded processor need a large amount of time to capture and process the sequence of images. Using FPGA, it is possible to parallelize the algorithm processing and capture with a double-buffer technique. The system proposed is based on a field-programmable gate array (FPGA) with a synthesized microcontroller.

The first step of our algorithm is to capture at least M images from the scene without objects to detect. Then, the background model is obtained, which will serve as a comparative element to determine the presence of people or objects in the scene. Once the background is generated, in real time, new images are captured and introduced into the system to detect the presence of people. Detection is the comparison result between the background and the current image. Next, the edges of objects in the foreground area previously detected are obtained via an edge filter mask (horizontally and vertically) and then summed up. The circular edges correspond to head candidates. This is made via an annular filter bank that takes

into account the deformed ellipsoidal edge of a head. After this step, the background must be updated with the pixels of the current image without objects, and the data of head candidates are sent to a Microblaze to track and count people.

Shengsheng Yu et al. (17) also used a vertical camera and they proposed a new algorithm of detecting for moving people based on edge detection that is independent from some specific video clips and that has good performance in real time. Therefore, they construct a foreground/background edge model (FBEM) rather than the Gaussian Mixture Model that consumed too much CPU resources, which make the algorithm can not serve the real-time task. The canny algorithm is applied to obtain edge information from every frame in video clips. The result is a threshold image after canny algorithm, and the pixels those mean edge are white, other pixels are black. However, it can not distinguish that one edge pixel belongs to background or foreground. So the task becomes to confirm the type of every edge pixel. Their idea is that in most of the time, background images and its edges are static while foreground images and its edges are moving. Therefore, they can count the number of being edge (white color) for every pixel in consecutive frames. If the number is close to total frame number, it belongs to background; if the number is small, it belongs to foreground when it is white color in some frames. They also propose to use two relatively simple methods to achieve people tracking that are good for the real-time. The first method is to match the mass centers of moving people regions in previous frame to the new mass centers detected in current frame. Since every two persons that are close to each other within a threshold may be not necessarily a successful match, they take the similarity of the size ratio of moving people as the second method. This check is motivated by the fact that the size of moving people do not shrink too much between consecutive frames.

Mario Vento et al. (18) present an innovative method for counting people from zenithal mounted cameras. The proposed method is designed to be computationally efficient and able to provide accurate counting under different realistic conditions.

The method can operate with traditional surveillance cameras or with depth imaging sensors. In order to satisfy the efficiency constraint, their method, in contrast to the classical approaches, does not require the people detection and tracking steps. The foreground detection is performed using a quick background subtraction color-based algorithm. This feature makes the algorithm very attractive in a real-time context such as the people counting. On the other hand, this choice requires the design of a smart sensor able to effectively perform the counting task using only the information provided by the background subtraction step.
The proposed sensor is geometrically characterized by a rectangular area, with a crossing direction associated to it. It is possible define the width of the sensor W, as the side of the rectangle perpendicular to the crossing direction and the height of the sensor H, as the side parallel to the crossing direction. The rectangle is divided widthwise into K stripes with a width $\delta = W/K$. Each stripe is divided in two cells with a height H/2. The algorithm stores the current activation value and the previous activation value A for each cell, while for each stripes it maintains an activation value initialized to 0. The evaluation of the activation sequence is used to detect separately, with a good reliability, people walking with carts or in a queue in both directions. This single condition is not sufficient to properly solve the problem of counting people walking nearby in the same direction. For this reason, the algorithm searches contiguous sequences of cells that have an activation value equal to 1. The algorithm disables a stripe only when also its adjacent are deactivated and counts a number of persons that corresponds.

Regarding the depth sensors, in the framework of Vera, Zenteno, and Salas (19) they detect people using a Support Vector Machine classifier, follow their trajectory by modeling the problem of matching observations between frames as a bipartite graph, and determine the direction of their motion with a bi-directional classifier. They use zenithal depth camera. To use the height as a descriptor of the objects they observe, they normalize their measurements and express them with respect to the scenario floor, which is assumed flat. To that end, we rotated the 3D points computed from the depth images in order that the z-axis would coincide with the

floor's normal orientation. Their people detector is based on an application of Dalal and Triggs' method of the histograms of oriented gradients (20) but they have adapted this methodology to detect people using zenithal depth cameras. They define a standard size bounding box for people detection of 96 × 96 pixels. The size was chosen by calculating an average over 993 true positive samples and approaching the resulting size to suitable cell and block elements. The features were extracted by computing the gradient over cell structures; the orientation of the gradient was clustered into nine-bin histograms; a person is described by a feature vector x of size 1089 × 1. For the tracking, they model the problem of matching observations in frame j with observations in frame j+1 as a complete bipartite weighted graph G . The vertices $O = \{O_j, O_{j+1}\}$ are divided between the observations made in the frame j and the one that follows it. The set of edges E represents the hypothesis that two particular observations correspond to the same person. At the end, with the bi-directional classifier they want to classify a person's trajectory as going in one of two opposite directions.

Huiyuan Fu et al. (21) propose a new algorithm by multimodal joint information processing for crowd counting. In their method, they use color and depth information together with an ordinary depth camera. Specifically, they first detect each head of the passing or still person in the surveillance region with adaptive modulation ability to varying scenes on depth information. Then, they track and count each detected head on color information. The characteristic advantage of this algorithm is that it is scene adaptive, which means the algorithm can be applied into all kinds of different scenes directly without additional conditions.

They propose a color gradient model to define the depth effective graph. The advantage of this model is that it can represent the distance to the camera more visually and more semantically. They use three colors (red, green and blue) to label the corresponding practical locations of the different colors. It is possible to find that the color of floor (0 cm) is pure blue, and the color will gradually transit from pure blue to pure green at the range of 0–90 cm. Moreover, the color will gradually transit from pure green to pure red at the range of 90–180 cm. Using this color gradient model, it is possible to calculate the rough locations of humans in their

captured region according to the shown colors directly. For the extraction of the heads, they know that the peak of a head is closest to camera, so the depth data of head peak is maximum. For the same reason, the foot of person is far from the camera, so the depth data of the foot is minimum. From the top view of camera, they can always find the head peak comparing to other parts of body. To overcome the difficulty that each person has different height, they extract each head region based on former scene-adaptive scheme facing different scenes. Through extensive observations of depth graph using the color gradient model representation, it is possible find that each person on the floor can appear a peak when the floor is regarded as zero layer. In the very crowd scene, each head of person can still keep this peak. After above head detection step in depth domain, they transform to process in RGB domain from contour detection; they do not use traditional Kalman filter and particle filter based tracking method in the algorithm. Instead, they present a mixture model based tracking method fusing the match between frames comparing with position, shape size and distance.

Daichi Kouno et al. (22), in their method, extract four features from each image; body height, body dimensions, body size and depth histogram. Then they apply these four features into the AdaBoost algorithm. They use the Kinect as depth overhead camera and from that, they have person's area extraction, then the feature extraction and, at the end, a classifier for person identification. Their final goal is to identify the people in the images.

This work of thesis inspired by a part of work of Xucong Zhang et al. (23). They use the Kinect camera in overhead position in order to obtain the detection and the counting of the heads. Since the head is always closer to the Kinect sensor than other parts of the body, people counting task equals to find the suitable local minimum regions. They propose a novel algorithm that can effectively find local minimum regions with the advantage of locality, scale-invariance and robustness. Their algorithm is motivated by the water filling process, that the water moves away from the heave and out to the nearby hollow under the force of gravity until the

gravitational potential energy can not be reduced any more. They simulate the rain by generating the raindrop according to a uniform distribution. Once a raindrop arrives, they compare its landing spot with its neighborhood and find the descent direction until it can not descend any more, then the number of raindrop at the balance spot increases. They take the depth image as a function f, where f(x, y) stands for the depth information of pixel (x, y). Due to the noise of Kinect sensor, f(x, y) can be non-derivable or even discontinuous. Finding people in depth image equals to finding local minimum regions in f. They introduce an additional measure function g(x, y) to "measure" f(x, y).

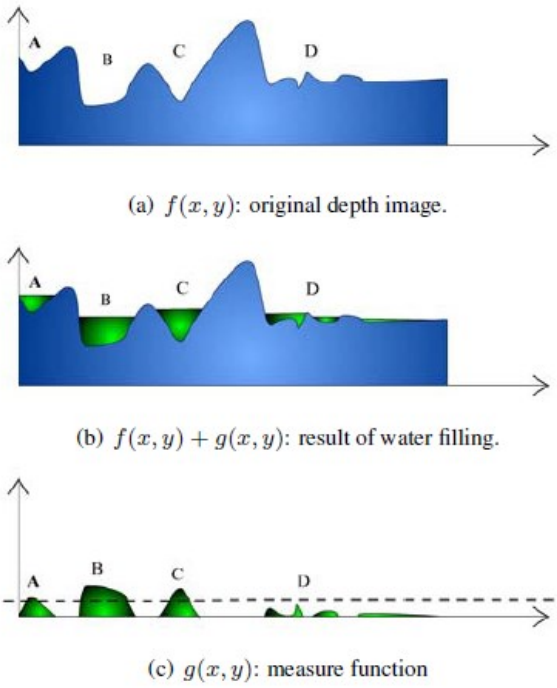


Figure 6 – Water filling example

With this technique, it is possible to obtain the heads and so then, they add a tracking module to validate the counting performance.

This thesis is inspired by this method in the part of the water filling process simulation, thinking therefore to a depth image as a territory to be flooded.

3 Description of the method

In this work, therefore, to solve the occlusion problem, we used the overhead Kinect camera and so, to find the heads of the people, which are the minimum local regions in this view, are used the depth images as one input of the algorithm. The other input is a configuration file, in which there are some values of fundamental characteristic parameters of the algorithm. The idea is that a user could change these values depending on the context of application of the algorithm.

The algorithm proposed, which allows getting excellent results, consists of many stages. In particular, it is possible to identify them in this way:

- Pre-processing;
- Water Filling;
- Post-processing;
- Tracking;
- Height estimation;
- People Counting.

In the first phase, we work on the original depth image provided by the camera. In order to reduce the noise, are used some morphological operations. After the application of the Gaussian Blur, there is the most important operation of this step: a threshold based on the distance. The idea is that everything is very close to the floor is not important for the final result.

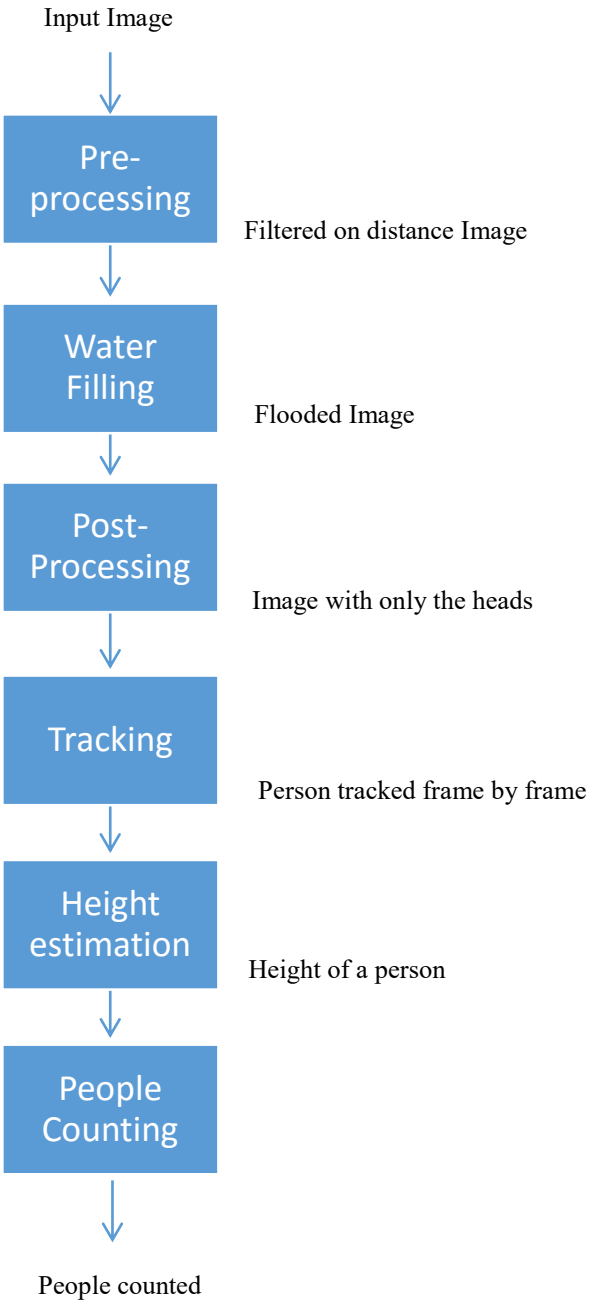


Figure 7 – High-level architecture

In the images below, we can see the use of the filter in the first step. The first, is the original image, and then, in the second, it is possible to see how some parts of it has been filtered (legs and feet especially for example).

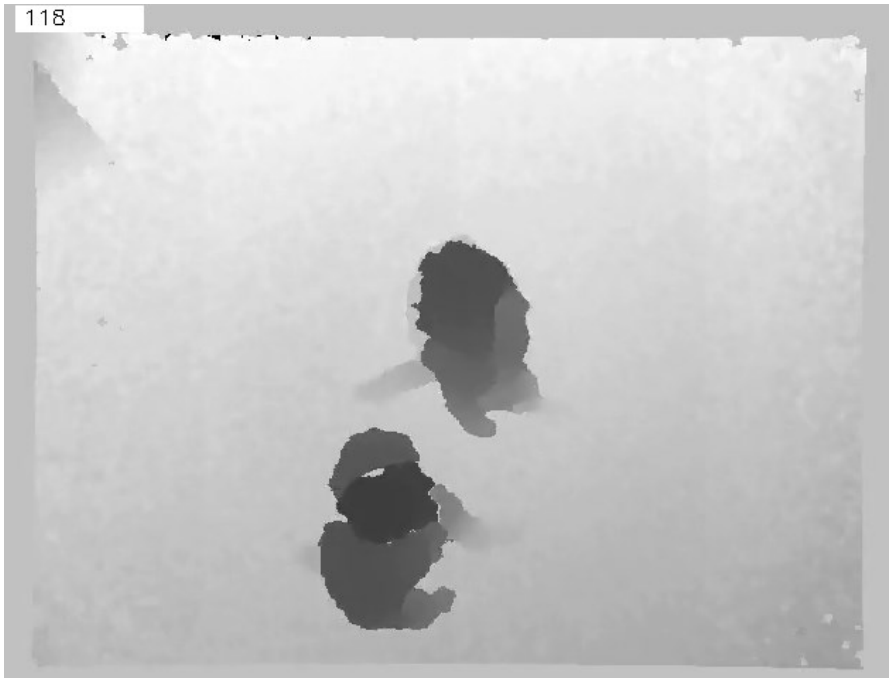


Figure 8 – Original image before the threshold

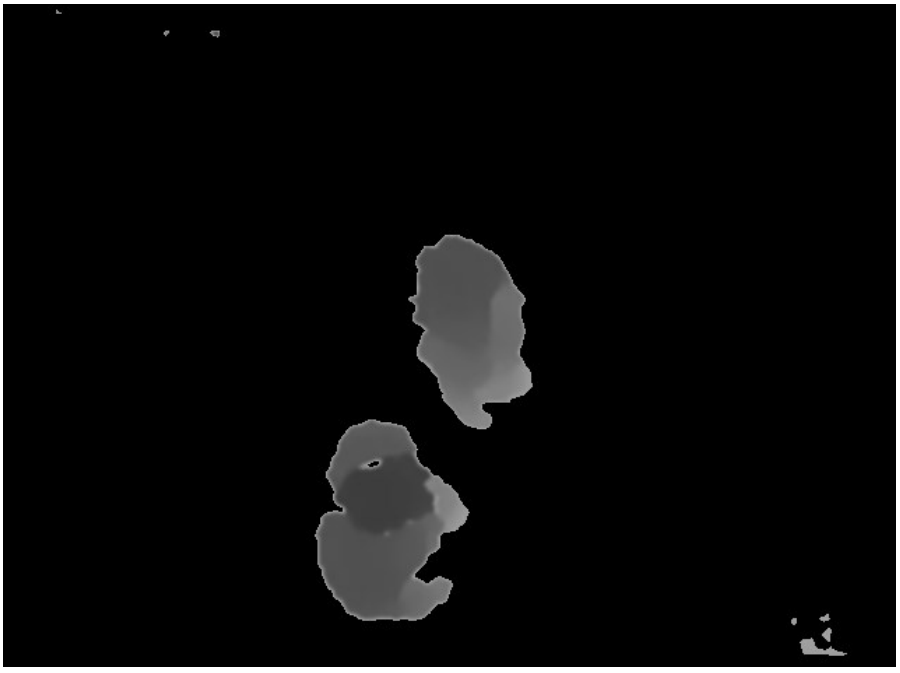


Figure 9 – Filtered image height

On this second image, we can apply the Water Filling process. Considering the image as a land, this algorithm goes to simulate a water flood. Using the images below, it is easier to understand how we find the heads.

Taking the depth map as a land, with humps and hollows, the value of depth represent the height, just like the picture.

5	5	4	5	6	6
4	4	4	4	5	6
2	3	3	3	4	5
2	2	3	3	4	5
1	0	2	2	4	6
1	1	2	2	6	6

data of a depth image

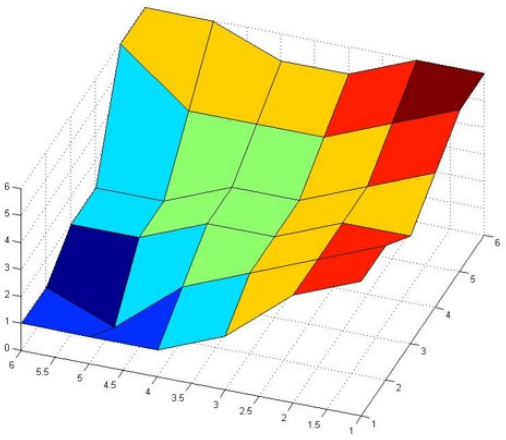


Figure 10 – How Water Filling works (1)

Then, assume the green point is a raindrop, and the first location on the land is where we see it. The raindrop will flow directly to the neighborhood hollow under force of gravity. Therefore, we compare the value of the green point with its neighbor eight points: if the green point get the lowest value, the drop will stay there, no more move, else we will find the lowest value of its eight neighbors, and move the drop to there.

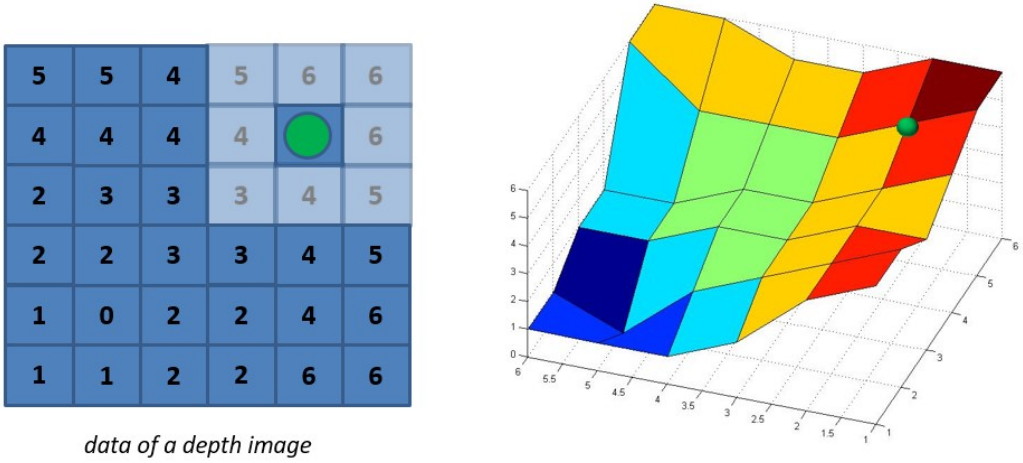


Figure 11 – How Water Filling works (2)

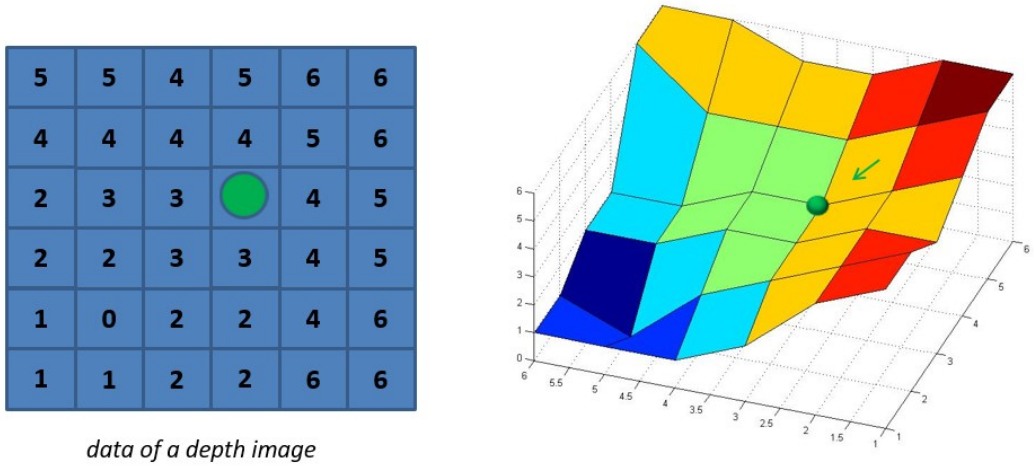


Figure 12 – How Water Filling works (3)

In the next image, there is the next compare and it is very important: the drop will follow the inertia**,** which means the last chosen direction will be a prior direction; if there are several equal lowest values among the neighbor points, just like this case, the drops will prior to choose this direction to move. That will help the drop to move through the plane region faster.

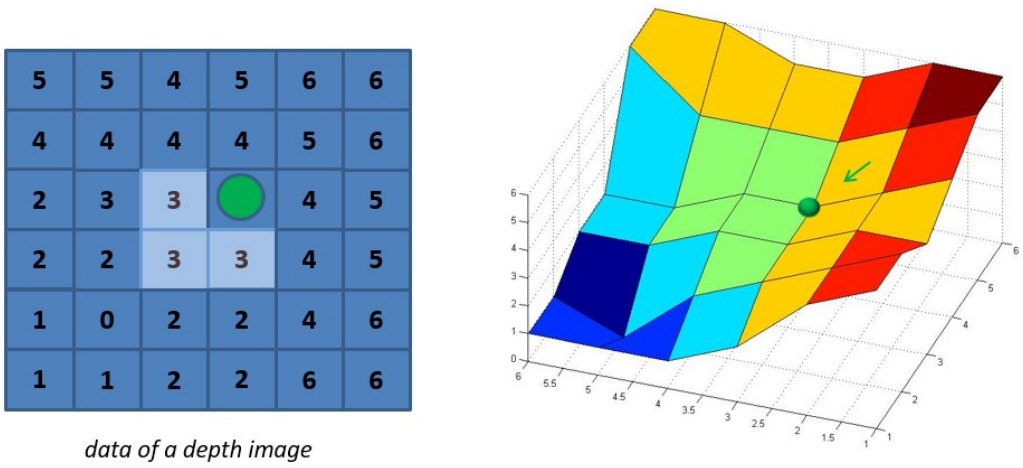


Figure 13 – How Water Filling works (4)

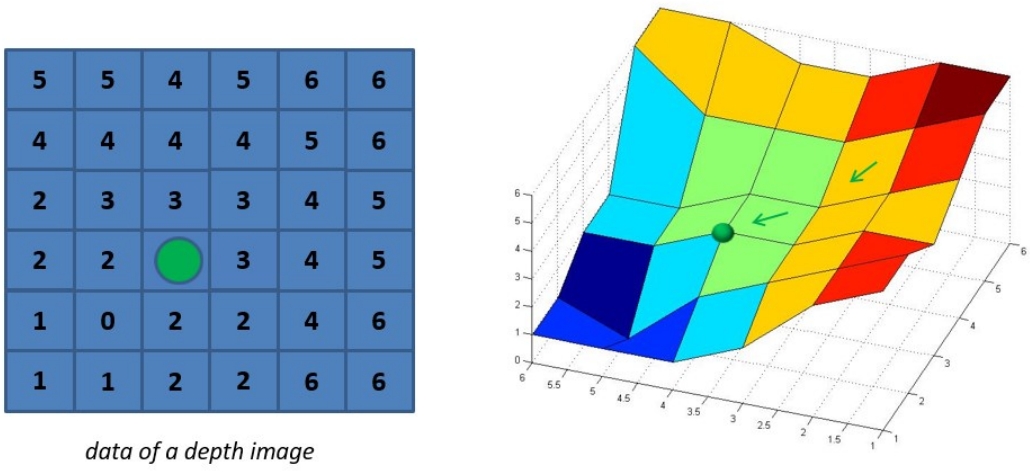


Figure 14 – How Water Filling works (5)

On the next move, the drop find the lowest position to stay, and the value of the point will plus one. Just like this in the example, which changes from 0 to 1.

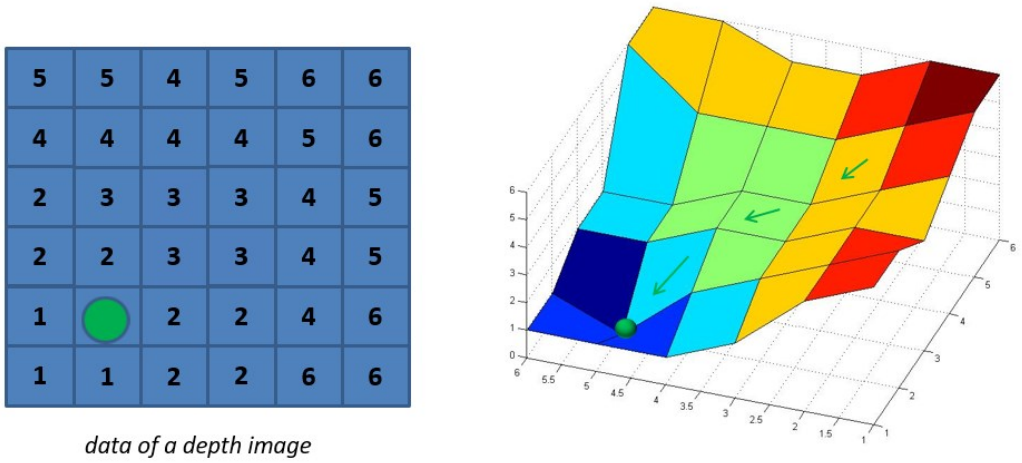


Figure 15 – How Water Filling works (6)

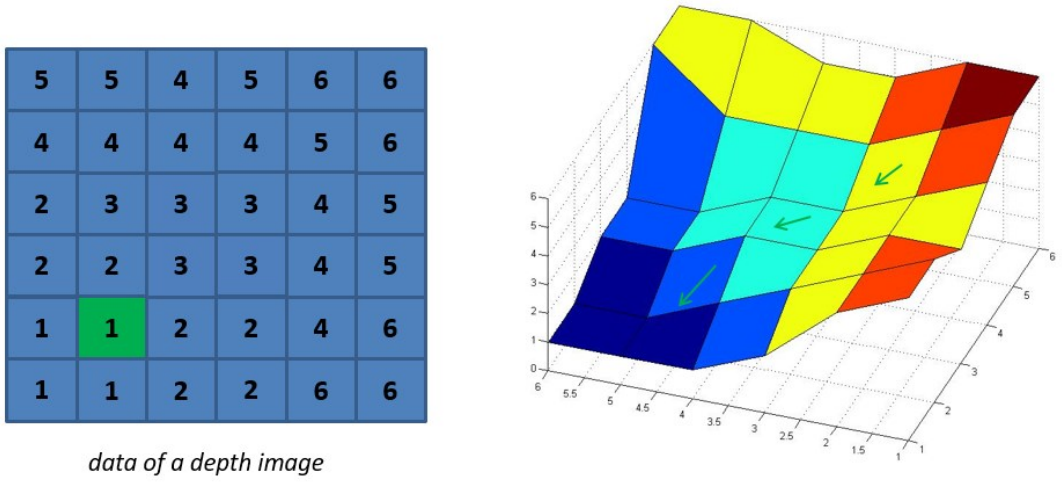


Figure 16 – How Water Filling works (7)

Now we can imagine the profile of a person in a depth image. If we think about an overturned image, we can apply the same procedure as above: the raindrops falling from the sky, they arrive on the shoulders and then down into the head.

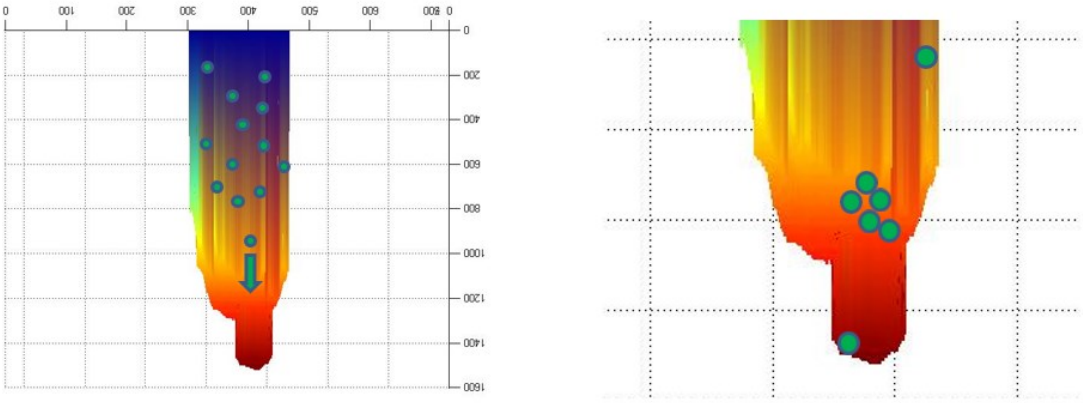


Figure 17 – Application of Water Filling with a person

In a complete depth image, it is possible to have many elements. After the flood, it could be necessary to filter based on water levels. For example, we can see the image below. Here is the floor, the people and other noise that can not be eliminate. For us the important is only to know if in the image there is one or more people: we can filter out them, based on how many levels of water are present in the head.

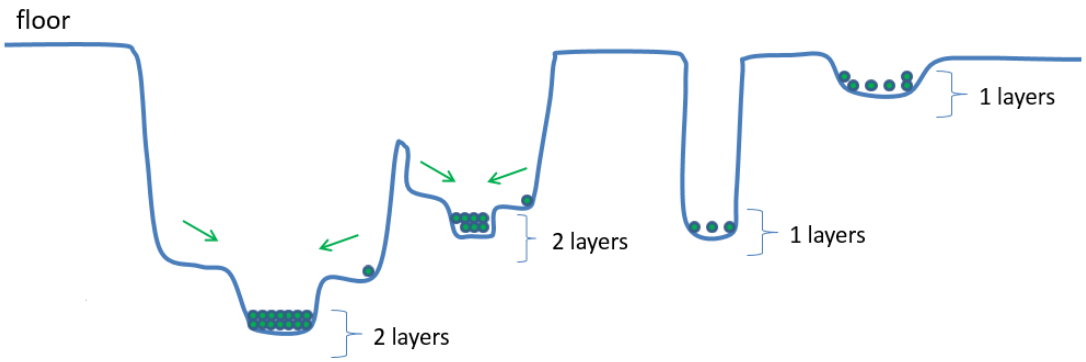


Figure 18 – Example of filtering the water level

In fact, in the first step of water filling, the drops will drop down from the sky to the land, and then they will find their way to the lowest place just like the rain drops under the force of gravity. Nevertheless, very important, the head will get the drops from the shoulder, but the noise will not, so at last the drops on the head region will get more layers than the noise.

The proportion between the head and shoulder is almost static, which means the big head with bigger shoulder, so the drops layers of the head is almost the same, so no matter it is child or adult, and no matter the height of the camera, the head will always get the same layers of drops.

This method can effectively find local minimum regions with the advantage of locality, scale-invariance and robustness.

The next image shows the real application of this algorithm in this proposed work.

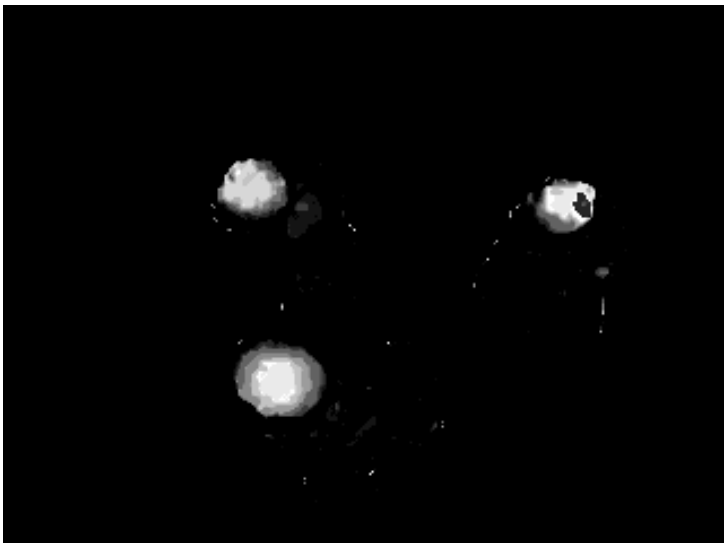


Figure 19 – The application of Water Filling

The third phase consists in some operation of post-processing: in particular, the Opening operation (it is obtained by the erosion of an image followed by a dilation)

is carried out with the aim to filter the noise and to leave on the scene only the heads. This sequence of operations in fact is useful for removing small objects from the scene.
The kernels sizes used for the erosion and the dilation make part in the parameters of configuration of the input file and so the user can decide them.

The fourth step is represented by the use of the tracking module that is very important for the last phase, that of people counting, and, therefore, for the detection of crossing direction. That is how it works: the module associates a person with an object to track and, farther, it verify that each person, each blob representing the head, is associated with an object for each frame. It checks if in the scene there are new blobs that are really different from the blobs in the previous frame or if they are the same but which have moved.
Each object is associated with an ID in order to be identified, and with a rectangle that include it.

After the detection, in the fifth step, it is possible to calculate the height of each person, using the information of the distance between the camera and the head of each the person, that was obtained in the previous steps. Exploiting the information of the depth images, it is possible to work on the pixels in the region of the rectangles that include the heads. Obviously, the only important pixels in the rectangles are the pixels that represent the heads. The average between the values of the pixels and the number of pixels of interest means the distance from the camera. The information about the height of the camera is a configurable value present in the input file, because depends on the situation and it can change depending on the context.
At the end, making the difference between the height of the camera and the value of that average, it is possible to get the height of the examined person.

In Figure 20 an example, with the ID number and the height of the person, in meters.

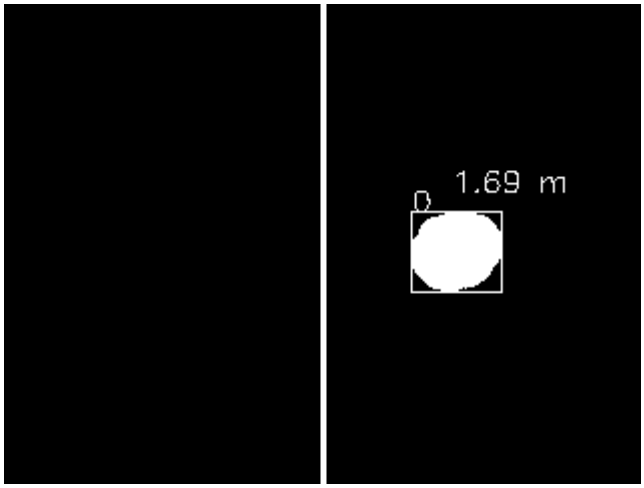
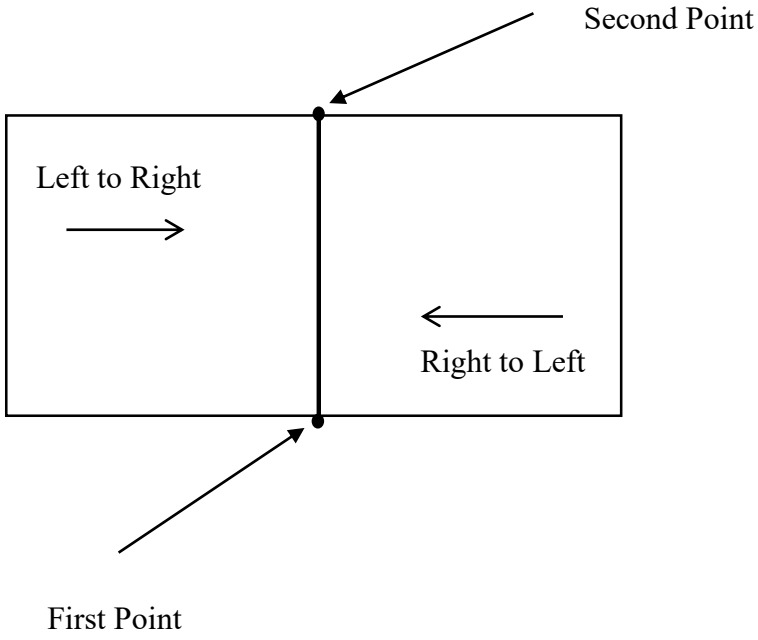


Figure 20 – Height Estimation

In the last phase of the algorithm, the final counting is carried out: the idea is to verify if the points of the rectangle associated with the object cross the line that identifies the sensor for the counting. Each person can cross this line in bidirectional way and each person is counted when at least three of the five points of the rectangle (four corners and the center) move from a side to the other side. When a person appears, she was awarded a state (-1) that denotes that she is a new person to count. Depending on the direction of crossing, then is assigned the relative corresponding new state, which is also important to determine if the same person passes again the sensor without leaving the scene.

The sensor for the counting can be placed in all of the ways that the user decides. It is designed so: it has two application points and the user decides which of the two is the first point and which is the second and where to place them. Depending on the user's choice is determined the direction of crossing. For example, in the case of a sensor placed in a vertical position with respect to the image (parallel to the axis of ordinates in the image), it is possible to decide what is the direction that represents the crossing from left to right and right to left, depending on the choice of points. In the following Figure 21, this concept is explained.



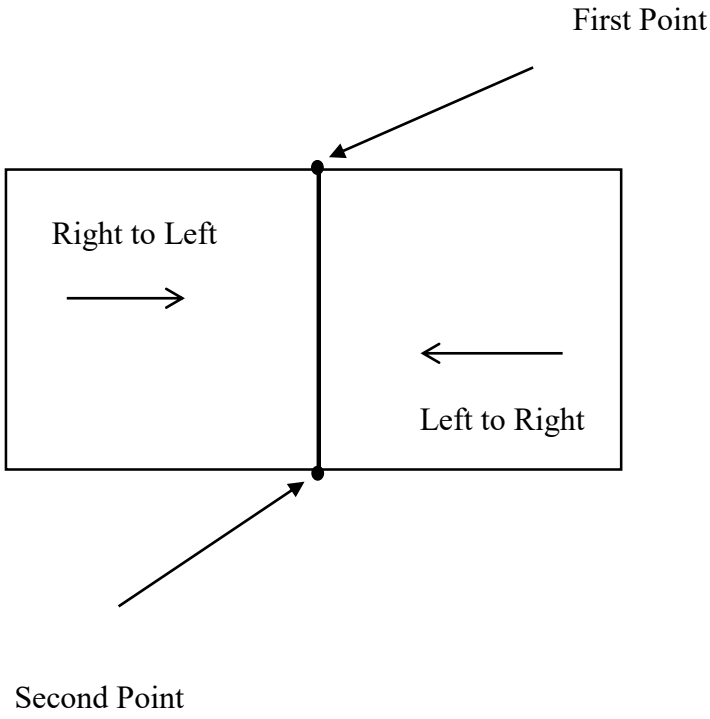


Figure 21 – Example of Counting Sensor

4 Experiments

The algorithm described in the previous chapter has been tested in all respects. In this chapter, in fact, there is an analysis of the dataset on which it was tested, the accuracy of the results and a profiling of the same algorithm.

The dataset used in this work is the same dataset presented and discuss in literature thanks to Mario Vento et al. (18).
In fact, the results of this thesis were compared with the Groundtruth produced on the same dataset but with their method.

4.1 Dataset

For the application of this work, we used the dataset mentioned above that it is possible to divide, for greater clarity, in two parts: Dataset$_C$ and Dataset$_U$. For the first type of these datasets, the subscript *C* stands for the term *Controlled*, meaning that the dataset was acquired in a controlled environment where each person who crossed the counting line had preliminarily received specific instructions regarding how to transit. On the other side, in order to assess performance of the method in a real scenario, in the second dataset persons freely flow across the virtual line and so the reason of the *U* for *Uncontrolled*.

For the realization of the dataset, the camera is located at a height of 3.2 m from the ground in a zenithal position so as to frame the persons from overhead.

The first type of these datasets, Dataset$_C$, contains sequences acquired in two different scenarios: indoor and outdoor. Each scenario comprises sequences with

an increasing number of persons that flow within the area of interest in the same direction and/or in the opposite directions. In the simplest case, there is a single person that crosses the area framed by the camera, while in the most two complex cases there is a group of six persons which cross the area proceeding either in the same direction or in the two opposite directions.

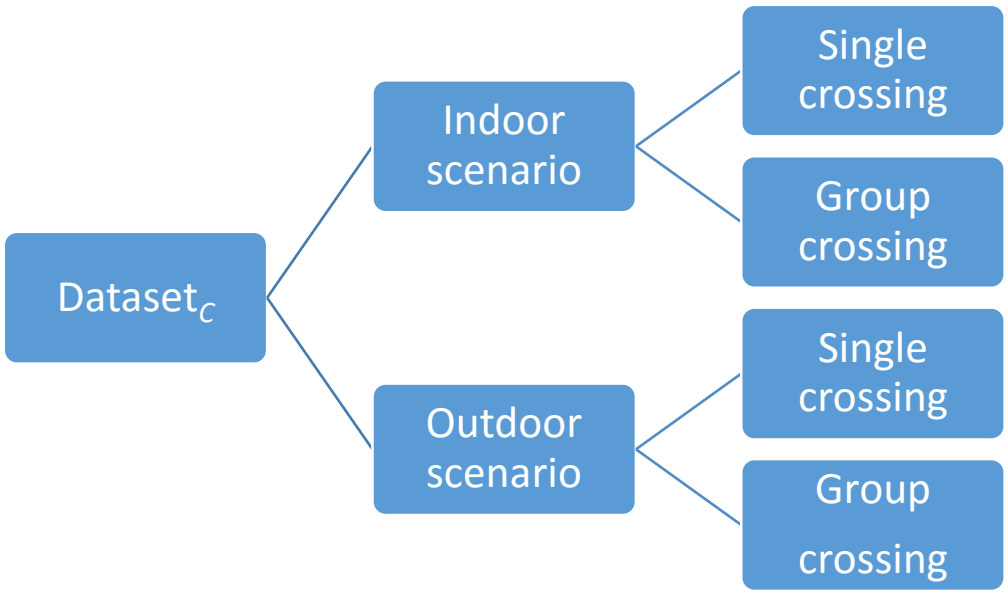


Figure 22 – Description of Dataset$_C$

The Dataset$_U$ is not composed in this way. It has only the indoor scenario but it is very different from the previous Dataset$_C$. It has been recorded at the entrance of a corridor in uncontrolled way, thus allowing the free flow of the persons under the camera in both directions. This allowed collecting several complex situations, which are not present in the first dataset and may typically arise in real world scenarios, where people may cross the virtual line in unconventional ways.

In this second part of dataset, it is possible to see that the camera is located very close to a door. Furthermore, there are mixed situations as regards the crossing of persons: it is possible to find a single crossing after a group and vice versa, one or more persons standing in the area of interest, persons that cross diagonally, persons

that suddenly stop and re-start, persons crossing with medium/large-sized objects or very dense groups of persons passing through the passage (not only six persons).

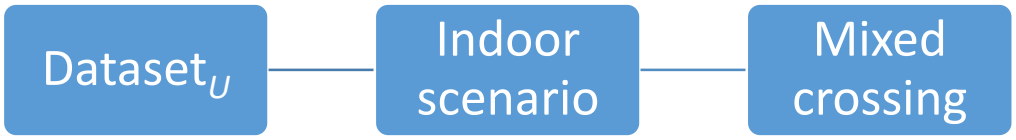


Figure 23 – Description of Dataset$_U$

In the Dataset$_C$, we have positioned the counting sensor in the middle of the scene, according to the previous analysis made on it with the other scientific work. At the same time, for the Dataset$_U$, the sensor has been positioned more close to the door, on the left side of the scene.

In the figures below, it is possible to see some examples taken from the complete dataset. Both for the Dataset$_C$ and for the Dataset$_U$, we have traditional images (RGB camera) and the depth images (Microsoft Kinect camera), used for the final purpose.

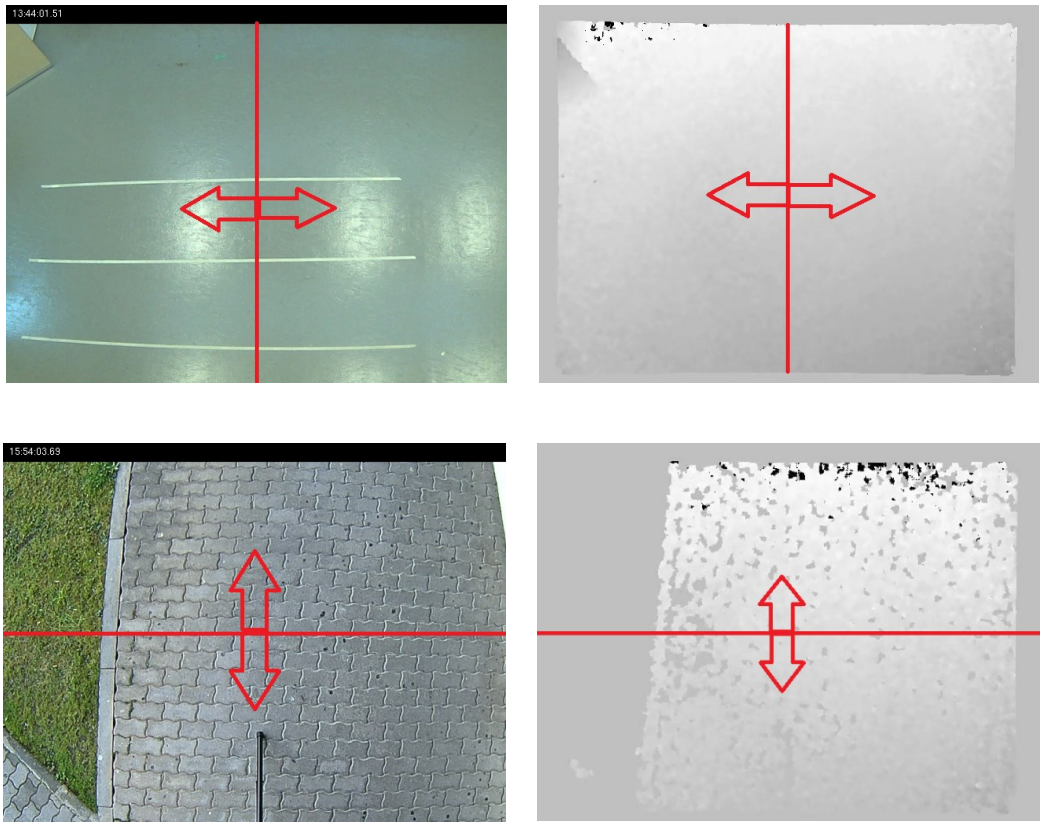


Figure 24 – Example of indoor and outdoor scenarios of Dataset$_C$

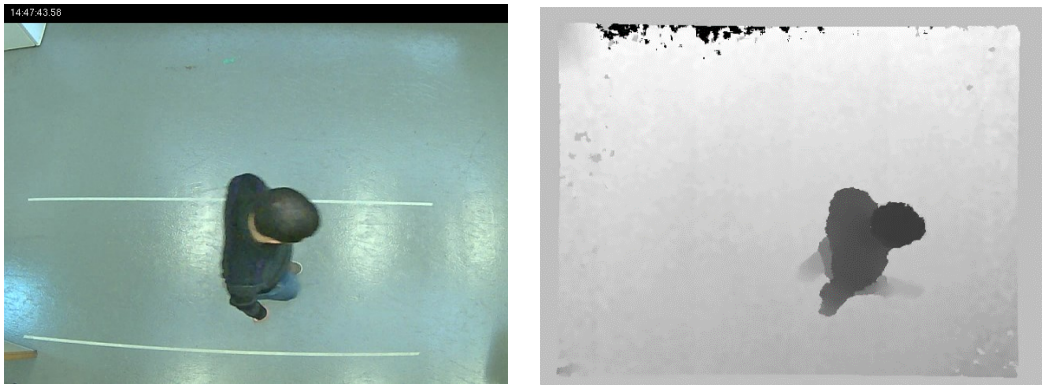


Figure 25 – Example of single person crossing in indoor scenario of Dataset$_C$

Figure 26 – Example of single person crossing in outdoor scenario of Dataset$_C$

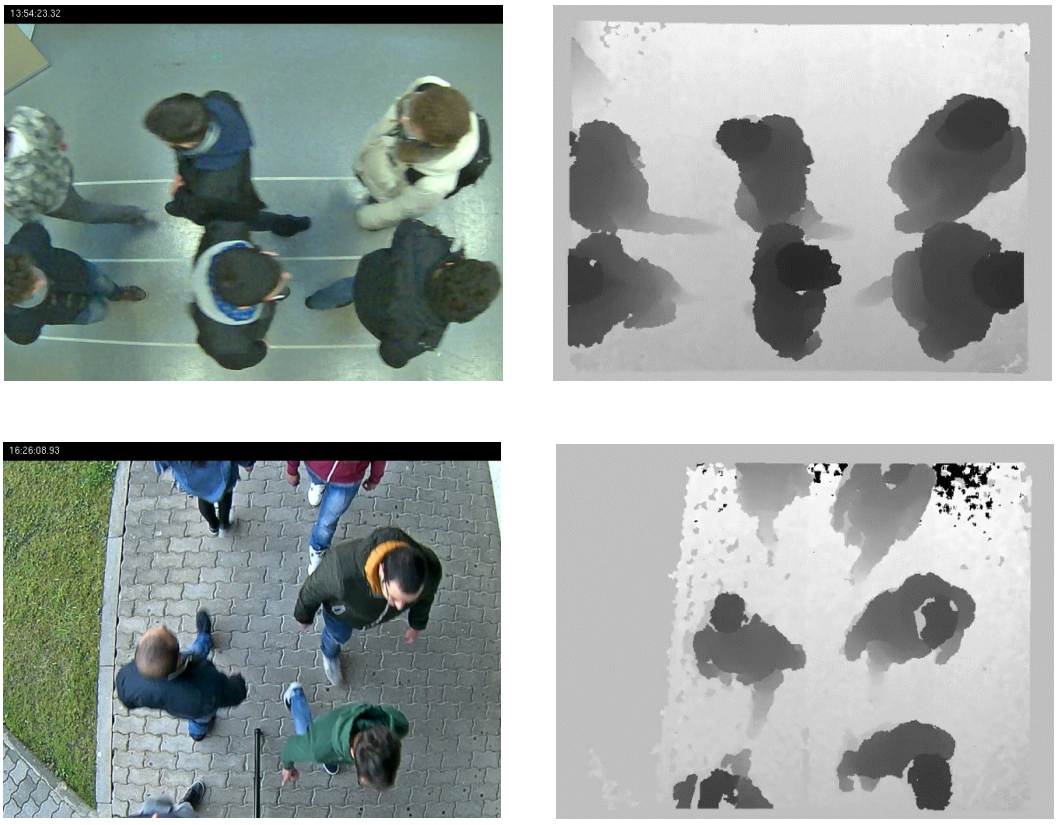


Figure 27 – Examples of indoor and outdoor groups crossing of Dataset$_C$

Figure 28 – Examples of crossing in Dataset$_U$

This dataset is very useful to test an algorithm like that. In fact, we have the counting when is prevalent a natural illumination (outdoor) or when the source of illumination is exclusively artificial (indoor). It is important to note that the two scenarios above were specifically designed to resemble typical real world environments where people counting systems are adopted, paying attention to the characteristics of the flooring. In fact, in the indoor scenario, in such cases, the illumination is assured by a distributed artificial lighting system with a relevant reflection from the flooring. On the contrary, in the outdoor scenario, these situations are generally characterized by the presence of a matte pavement (no reflections) and indirect but prevalent sun lighting.

In the other hand, it is useful also because the analysis is made considering also the different crowding conditions in both datasets.

4.2 Accuracy analysis

In this paragraph, we report the results of the performance assessment of the proposed people counting method on the previous dataset. The first thing to do, however, is to describe the parameters with which the algorithm was evaluated for the performance.

The performance are measured in terms of:

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $f - index = \frac{2*Precision*Recall}{Precision+Recall}$

TP is the number of true positives, for example transits of persons that are correctly detected by the method; FP is the number of false positives, for example falsely detected passages of persons; FN is the number of false negatives, for example passages of persons missed by the method.

The results in terms of these three previous values are shown in the tables below.

Dataset	Scenario	Flow density	TP	FN	FP	Precision	Recall	f-index
Dataset$_C$	Indoor	Isolated transit	212	0	0	1.0	1.0	1.0
		Group of persons	203	0	4	0.98	1.0	0.99
	Outdoor	Isolated transit	222	14	0	1.0	0.94	0.97
		Group of persons	306	11	0	1.0	0.97	0.98
Dataset$_U$	Indoor	Mixed transit	913	22	25	0.97	0.98	0.98

Table 1 – Table with specific performance

How it is possible to see in Table 1, where the results are shown for each category, this method has excellent results. The best results are in the situation of indoor scenario with single crossing in Dataset$_C$: no error in any case, the maximum in precision, recall and f-index. This is the simplest case but the results are not always so good with other methods.
In the same scenario, but with group of persons, the results are almost as the previous ones.
Overall, for the Dataset$_C$ performance decreases only slightly for the outdoor event with a single crossing where we can see an FN number higher than the other cases.

For indoor case of the Dataset$_C$ in fact, the precision improves of about 2% and the f-index of about 1% passing from the group of person to the isolated transit case.
For the outdoor case, the recall and f-index improve of about 3% and 1% respectively, passing from the isolated crossing to the groups. This happens, probably, because the crossing occurs in the part of scene with less noise.

In the case of Dataset$_U$, we see a higher number of FN and FP considering, however, the complexity of the uncontrolled scenario. The presence of objects at considerable height leads to some errors in more than the other part of the dataset.

Also in Table 2, in fact, we can see that the number of FN increases in the outdoor case: this is because the sunlight affects negatively the quality of the depth image provided by the Kinect sensor, by increasing the noise in the input image that reflects in the reliability of the blob detection process.

In the indoor scenario, we have more FP, but this is because the parameters of input of the algorithm in the configuration file are determined by choosing a trade-off to get good performance on all stages that compose it.
The recall increases of about 3% passing from outdoor scenario to indoor, instead the precision increases of about 2% passing from outdoor to indoor. The f-index differs of about 1% between the two cases.

	Scenario	TP	FN	FP	Precision	Recall	f-index
Dataset	**Indoor**	1328	22	29	0.98	0.98	0.98
	Outdoor	528	25	0	1.0	0.95	0.97

Table 2 – Results divided between indoor and outdoor

In the next Table 3, it is possible to see that the results on the complete dataset are very important and so, very good for all three parameters.

	TP	FN	FP	Precision	Recall	f-index
Dataset	1856	47	29	0.99	0.98	0.99

Table 3 – Total results on complete dataset

The results of this work are compared with the results of the work (18).
In Table 4, there is the comparison between the two methods, the proposed in this work and the method proposed in the paper (18), on the Dataset$_C$; furthermore, since they used two cameras in their method (RGB and depth), the first comparison is with the use of RGB camera.

Method	Sensor	Flow density	Precision	Recall	f-index
Proposed	**Depth**	**Isolated transit**	1.0	0.98	0.99
		Group of persons	0.99	0.98	0.99
Proposed in (18)	**RGB**	**Isolated transit**	0.99	0.98	0.99
		Group of persons	0.98	0.84	0.90

Table 4 – Comparison between two methods on Dataset$_C$ with different cameras

The results are very improved in this situation in almost all the cases of crossing. The most important improvements, passing from RGB to depth, are: 1% in precision in the isolated transit case and 2% for the groups, of about 17% in recall in the flow density of group of persons, 10% in f-index again in the flow density of group of persons.

In Table 5, there is the comparison, again on Dataset$_C$, but considering the use of the depth camera also in the other method.

Method	Sensor	Flow density	Precision	Recall	f-index
Proposed	**Depth**	**Isolated transit**	1.0	0.98	0.99
		Group of persons	0.99	0.98	0.99
Proposed in (18)	**Depth**	**Isolated transit**	1.0	0.98	0.99
		Group of persons	1.0	0.95	0.97

Table 5 – Comparison between two methods on Dataset$_C$ with same camera

The recall improves, with the proposed method, of about 5% in the crossing of groups of persons. The f-index improves of about 2% in the case of groups of persons.

In Table 6, it is possible to see the comparison between the two methods on Dataset$_C$, but considering the two different scenarios of application and the two different used sensor.

Passing from RGB to depth in the different method, a great improvement in the recall is possible to see analyzing both scenarios: the recall improves of about 14% in the indoor scenario and 3% in the outdoor. The precision improves of about 1% in both scenarios, the f-index values improves of about 4% in indoor scenario and of about 2% in outdoor scenario.

Method	Sensor	Scenario	Precision	Recall	f-index
Proposed	**Depth**	**Indoor**	0.99	1.0	0.99
		Outdoor	1.0	0.95	0.97
Proposed in (18)	**RGB**	**Indoor**	0.98	0.88	0.93
		Outdoor	0.99	0.92	0.95

Table 6 – Comparison between two methods on Dataset$_C$ on two scenarios with different cameras

In Table 7 there is the same comparison of above, but considering the use of the same camera in both cases. Considering the division based on these scenarios of Dataset$_C$, the results are very similar. There is an improvement using the proposed work, in the recall in the indoor scenario of about 2%.

As regards the Dataset$_U$, which is only indoor but with various types of crossing, the comparison is shown in the Table 8. It is possible to see the difference of the results when the authors of the other method use the camera RGB. Passing from the RGB to depth camera, the precision increases of about 7%, the recall increases of about 15% and the f-index improves of about 11%.

Method	Sensor	Scenario	Precision	Recall	f-index
Proposed	**Depth**	**Indoor**	0.99	1.0	0.99
		Outdoor	1.0	0.95	0.97
Proposed in (18)	**Depth**	**Indoor**	1.0	0.98	0.99
		Outdoor	1.0	0.95	0.97

Table 7 – Comparison between two methods on Dataset$_C$ on two scenarios with same camera

Method	Sensor	Precision	Recall	f-index
Proposed	**Depth**	0.97	0.98	0.98
Proposed in (18)	**RGB**	0.91	0.85	0.88

Table 8 – Comparison between two methods on Dataset$_U$ with different cameras

Method	Sensor	Precision	Recall	f-index
Proposed	**Depth**	0.97	0.98	0.98
Proposed in (18)	**Depth**	0.98	0.92	0.95

Table 9 – Comparison between two methods on Dataset$_U$ with same camera

The Table 9 above shows the results on Dataset$_U$ of the two methods, using the same depth camera. The three values are generally higher; in particular, we have an improvement in the recall of about 7% and in the f-index of about 3%.

Table 10 shows the results of the two methods on the complete Dataset (Dataset$_C \vee$ Dataset$_U$) using different sensors: passing from RGB to depth sensor all the values improve. In particular, the precision of about 3%, the recall of about 11% and f-index of about 8%.

Dataset	Method	Sensor	Precision	Recall	f-index
DatasetC $\vee$ DatasetU	**Proposed**	**Depth**	0.99	0.98	0.99
	Proposed in (18)	**RGB**	0.96	0.88	0.92

Table 10 – Comparison between two methods on complete Dataset with different cameras

Also using the same camera, the results improve, as it is possible to see in Table 11. The precision is the same but the recall increases of about 3% and f-index of about 2%.

Dataset	Method	Sensor	Precision	Recall	f-index
DatasetC ∨ DatasetU	**Proposed**	**Depth**	0.99	0.98	0.99
	Proposed in (18)	**Depth**	0.99	0.95	0.97

Table 11 – Comparison between two methods on complete Dataset with same camera

The method proposed in this work leads to improvements in almost all cases compared with the mentioned method. Through all the presented tables, it is possible to see that this method is better. This is because in the (18), when they use the RGB camera, as described in the Chapter 2, they apply a completely different technique. The results compared to those obtained with that technology are much more positive and impressive.

Even if they used also the same depth camera, the results are still better on average: this is because, obviously, the technology used to count is different. With the depth camera, they cut the virtual scene at a fixed height before and they work on this new cut frame. To work in advance in this way on the image does not guarantee the management of all errors.

Another very important thing that makes the difference in the final results, is that in the comparison method they do not use a tracking module that is, instead, present in this proposed work.

4.3 Errors characterization

After the evaluation of the results, the characterization of the errors was the next step. Only with this characterization, it was possible to understand how to improve the method and solve some problems. The errors, which practically are FN and FP, are divided into macro classes with the aim of being able to improve them in the future.

In the next Table 12, there is the characterization of the False Negative that the algorithm has failed.

Merge of the heads	**Noisy images**	**Merge with other objects**	**Persons standing in the area of interest**
2	28	16	1

Table 12 – Characterization of False Negative

The Table 13 below instead, shows the characterization of the False Positive that wrongly the algorithm counted.

Accessories or parts of the body that are separate blobs from head	**Persons standing in the area of interest**
24	5

Table 13 – Characterization of False Positive

In the images below there are some examples of false negative and false positive characterized in order to understand how the algorithm could be improved. For each error, two images are shown that represent the same frame: the original frame provided by the depth camera and the image that show why the problem occurs in the phase of counting.

The first pair of images shows a false negative due to noise caused by light on a person. This is also, in part, due to the trade-off for the values in some function, which constitute some phase of the algorithm: in particular the morphological operation of erosion and dilation. The reason of the used values in this proposed work is that these values are a good compromise in terms of errors obtained and solved.

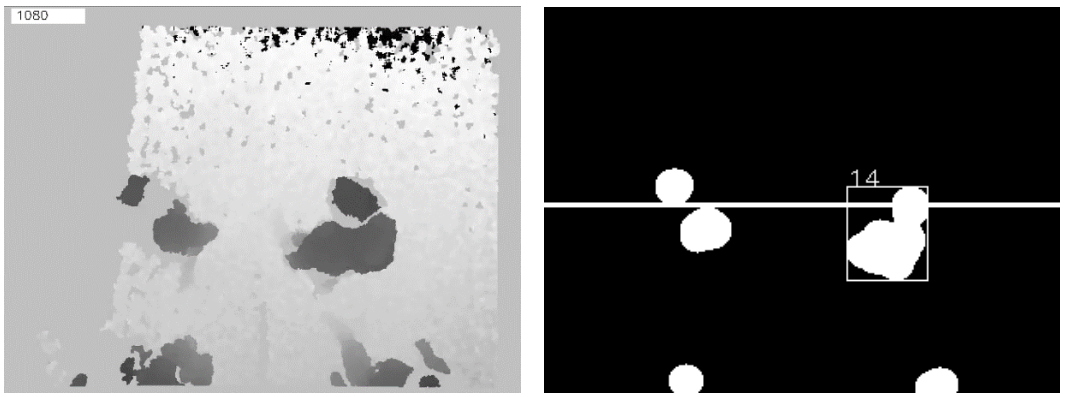


Figure 29 – False negative: noisy images (1)

In Figure 30 however, in the image provided by the depth camera, it is possible to see how the natural light produces a lot of noise in the head of the girl.

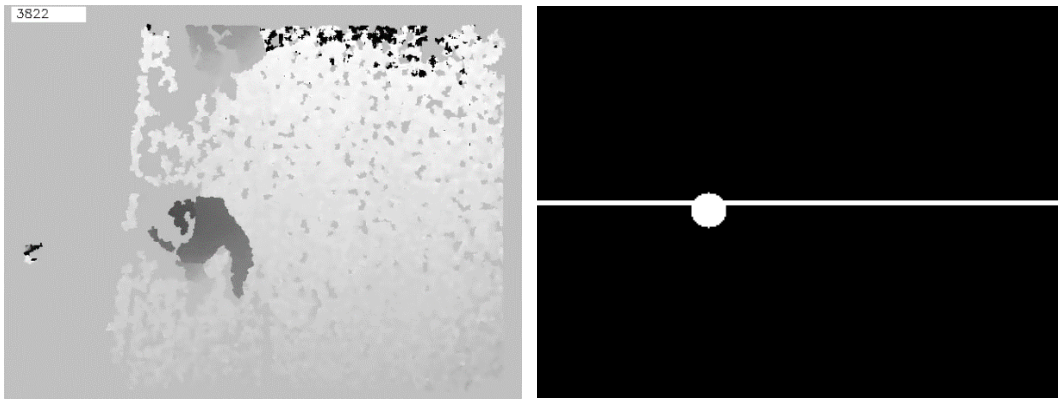


Figure 30 - False negative: noisy images (2)

This compromise creates a little problem that appears only twice: the merge of the heads of two persons that walk very close. In the Figure 31 below, it is possible to see an example.

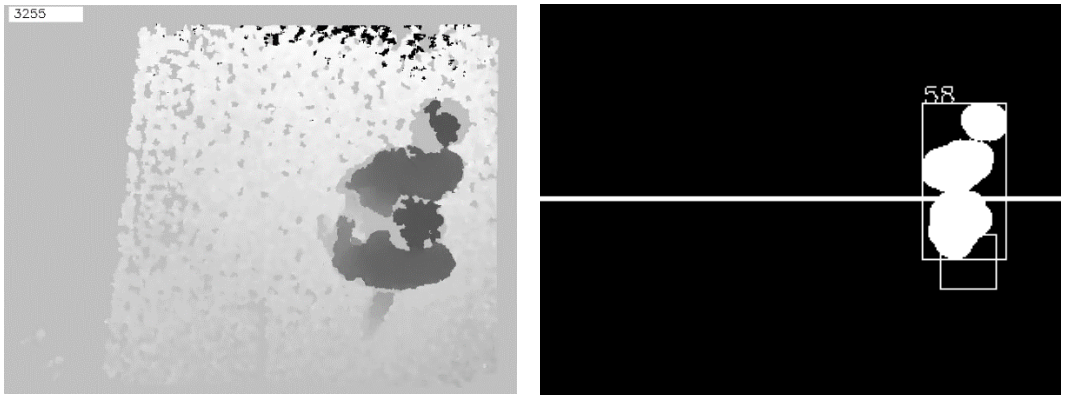


Figure 31 – False negative: merge of heads

In the Dataset$_U$ there is a fixed element in the scene: the door on the left, that is very height and sometimes it creates problems because the rectangle, associated to it after Water Filling, merges with the rectangle associated with a person. Because of this, we lose the information about the rectangle and ID associated with the person and the algorithm do not count him. In the images below an example.

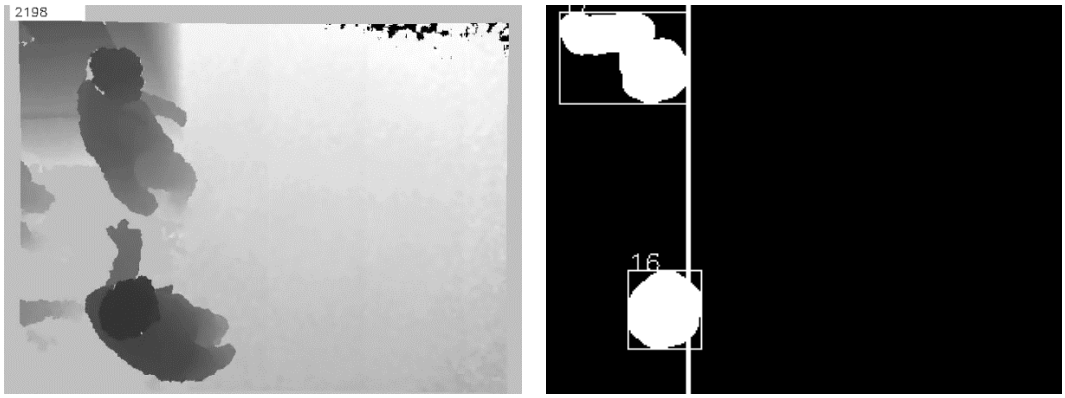


Figure 32 – False Negative: merge with other objects

The only case of person standing in the sensor area is a relative problem.
As said in the previous paragraph, this method was compared with another method present in literature. The person near the virtual line crosses the line but, in the verification of the script that compared the results, this happens not in the same frame interval present in the results of the other method. Therefore, the person with the ID 42, in reality, crosses the virtual line of sensor but only after a little time from when he came in scene.

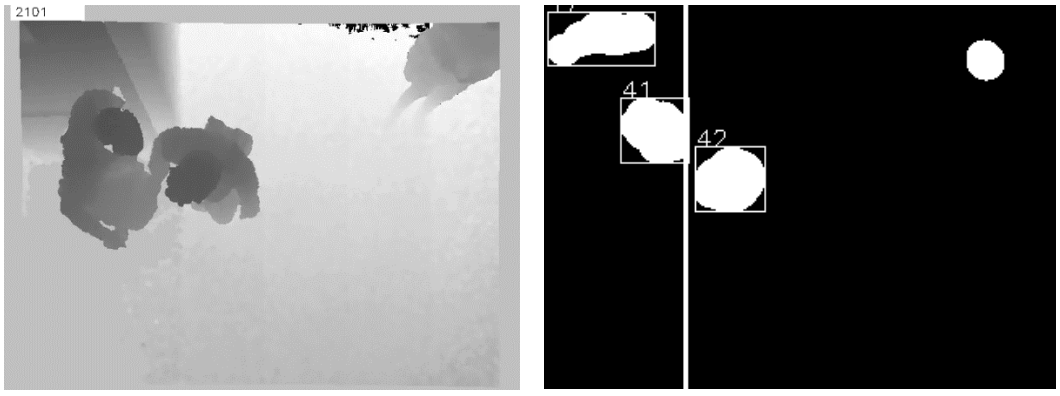


Figure 33 – False negative: person standing in proximity of sensor

As regards the false positive, it is possible to see some examples of the errors that belong to the two macro classes in the next images. In Figure 34 a problem with an object.

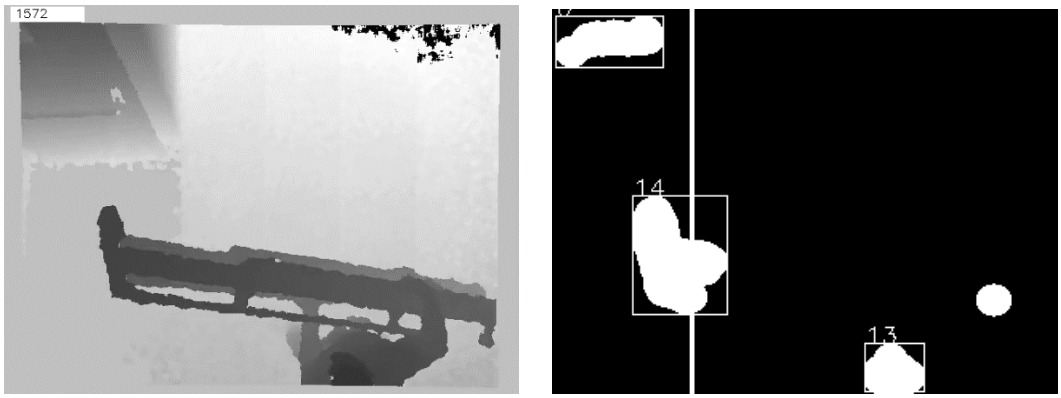


Figure 34 – False positive: accessories that are separate blobs from head

For the same reason of the false negative, but with the opposite results, a person standing in the proximity of sensor that crosses the line after a little time determines a false positive. In the Figure 35, the person with the ID number 21 spends the time in the sensor area before crossing it.

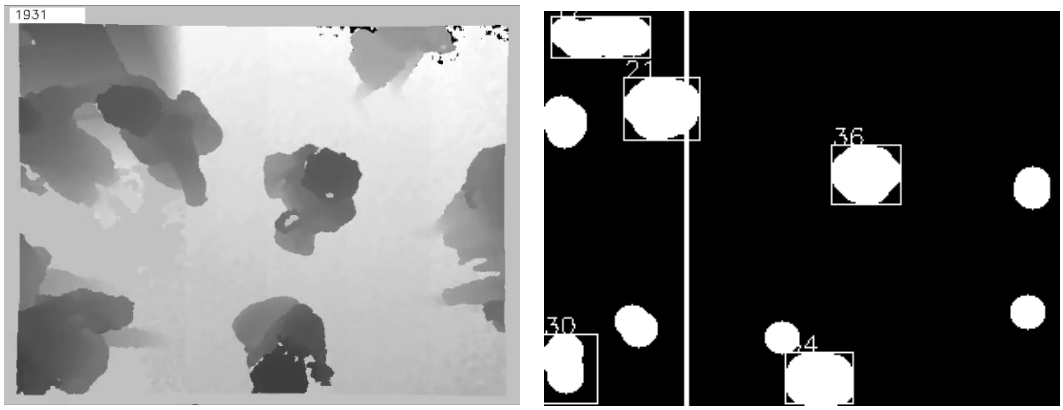


Figure 35 – False positive: person standing in proximity of sensor

4.4 Profiling

Another important analysis of this algorithm concerns the profiling. The aim of this study was to analyze the timing of execution of the algorithm, considering all calls to functions in it.
It was tested on a processing platform Intel® Core™ i5-3337U CPU @ 1.8 GHz with 6GB of RAM.

In order to analyze the worst cases of the dataset, it was made a profiling on the videos with the more complicated situations of crossing and with the highest number of people when possible, for example for the Dataset$_U$. For the same reason, in the Dataset$_C$ there is the worst case for the case of single crossing and crowded crossing, considering the indoor and outdoor scenarios.

In the Table 14, it is shown the time, in milliseconds, needed by the algorithm for each stage present in it. The values of each column represent the average considering each frame. If we indicate with C the current value of the duration of the step at the current frame, with B the number of frames and with A the value of the old timestamp until the current frame, we can write the formula of the final average in this way:

$$A = A + \frac{C}{B} - \frac{A}{B}$$

Therefore, in each final value we can see the weighted average on every frame that compose the video and on which the stage has ran. The last column is the sum of all the averages of the effected step, when the execution of the algorithm is terminated.

Name video	Pre Processing	Water Filling	Post Processing	Tracking	Height Estimation	Counting	Total
D_I_S	6.16712	12.876	2.87188	0.270522	~0	0.0219955	22.2075
D_I_G	6.04886	20.2327	2.79364	0.265682	~0	0.0363636	29.3773
D_O_S	6.47144	12.641	2.78365	0.237626	~0	0.0297872	22.1635
D_O_G	6.35442	16.5944	2.79904	0.247545	~0	0.0383649	26.0338
Depth_13_29	6.0443	55.0282	2.79664	0.644295	~0	0.0324385	64.5459

Table 14 – Profiling of algorithm

How it is possible to see in Table 14, it is possible to approximate to zero the phase of Height Estimation, because it is only composed by some mathematical calculation. The phases of Pre Processing, Post Processing and Counting, are, more or less, always constant for all videos. The values of Tracking and Water Filling instead, are higher in the last type of video, that of the Dataset$_U$. Since there are more persons in this last one, it is normal that the time of Tracking increases. More considerations must be made on the Water Filling.

The phase of Water Filling needs of more time in the cases of group of persons. The values, in fact, in the first four videos where *G* represents the groups, are slightly higher than the videos where *S* indicates the single person. Nevertheless, the video where this stage takes a long time is the last, where there are very crowded and uncontrolled crossing and situations of people standing. The reason is related to how the algorithm works.
In order to understand where it must fill with the water, it analyzes the entire image. For every blob that is not background, when it slides the image, it fills them with water, one at time sequentially. If we consider that this is the video with the most

crowded scenes, it is possible to understand why the time for this step increases so much. It must find a local minimum for each blob that it finds sliding the image, without knowing if someone is already met. Besides, in the last video, we already know that there is a tall door, which creates some problems.

The computational complexity of this step, indicating with n and m the variable associated to the dimension of the image, is $O(n * m)$. This is the best case, in which there are no blobs to fill. In the worst case instead, the computational complexity becomes $O(n * m * k)$ where k indicates the multiple of people present.

5 Conclusions and future works

In this thesis a very performant people counting system was presented. The results show that the method is very satisfying in different and real conditions of possible application. This algorithm is more performant than other method present in literature.
It has been tested on a significant dataset, present in literature and so already validated, of images that has been specifically devised and collected in order to account for the main issues that arise in typical installations, which may affect the counting accuracy. In particular, the acquisition technology (the depth sensor), the diversity of the scenario (indoor and outdoor) and the density of the people flow (isolated people and groups of persons).

As a future work, the performance may be improved by further expanding the dataset: for example, a larger dataset that should comprise sequences taken from other and real field installations coming from applicative domains, in different analytical conditions. It might be interesting to use a dataset where there is the presence of children, so as to exploit even better height characterization, even if in this there are also many smaller people than other.

The method could also be improved by looking for a solution to the errors obtained with this method.
For example in the case of the false negative and false positive. Considering the macro classes to which they belong, it would be possible to think of other solutions. It would be possible to think of using ROI as a possible solution to the problem of merging with other objects (especially in high places in the scene) that causes false negative. It would be possible to decide to change the input values of configuration

in order to modify the chosen trade-off that produces some false negative in the case of merge heads and noisy images, but considering that, important considerations were already made. Finally, it could work in order to lower the computational complexity on the Water Filling algorithm.

6 Acknowledgements

Chi leggerà questi ringraziamenti molto probabilmente mi conosce e sa, dunque, che non sono proprio il tipo da frasi strappa lacrime né tantomeno che scrive poi così tanto. Spero però che, tutto quello che magari non riesco ad esprimere scrivendo davanti ad uno schermo, lo possa esprimere e far arrivare attraverso parole e sguardi pieni di vita quotidiana condivisa.

Ringrazio la mia famiglia, rete sulla quale posso fare sempre affidamento e che quando penso che potrei cadere e farmi male, è sempre lì a proteggermi e salvarmi. Per i momenti belli vissuti insieme e che ancora vivremo. Grazie ai miei genitori, che mi hanno insegnato cosa vuol dire avere dei valori e che, ogni giorno, mi insegnano che bisogna affrontare tutto quello che ci viene posto davanti con serietà ma anche con serenità. Così come in tutta la mia vita fino ad ora, anche in questo percorso universitario il loro apporto è stato fondamentale, sia nei momenti belli che nei momenti brutti. Il loro esserci è per me essenziale. Grazie a mia sorella che, seppur lontana e con cui magari non capita spesso un confronto diretto, non manca mai di farmi sentire il suo affetto. Per me rappresenta un modello di riferimento per il coraggio e per il modo in cui decide di non abbattersi e di motivarsi, per la perseveranza nel voler ottenere quello che desidera. Ringrazio le mie nonne, da cui tutti potrebbero imparare ogni giorno che ciò che conta è saper accettare, con il piacere di andare avanti comunque, perché le persone che ci amano saranno comunque insieme a noi.

Ringrazio Valeria, per la sua passione, per il suo sapermi stare accanto, per la sua presenza, sempre costante e mai ingombrante. Nei giorni, nei mesi e negli anni di divertimento e di studio, in un periodo incasinato in cui sarebbe stato facile per tutti andarsene. Grazie per le scintille e per le incomprensioni, trasformatesi poi in nuovi rinnovamenti e promesse. Grazie per le gioie e per i momenti in cui non potrei desiderare altro. La dimostrazione che le cose belle arrivano all'improvviso.

Ringrazio tutti i miei amici, quelli che conosco da tanti anni, che non elencherò nome dopo nome, perché, per fortuna, molto numerosi. Un grazie per il tempo passato insieme, per le sere passate insieme, anche quelle dove si potrebbe stare a casa senza prendere freddo e pioggia e invece siamo lì, in quel luogo di incontro, che per tutti gli altri è semplicemente una piazza. Grazie perché continuiamo a essere numerosi, ognuno con i propri difetti e con i propri pregi: non è da tutti riuscire ad essere così tanti e così diversi, ma ad essere comunque insieme. Anche nei momenti di lontananza, una presenza che si è fatta sentire.

Ringrazio i miei compagni di classe, con cui, per un motivo o per un altro, non è sempre facile vedersi. Gli anni trascorsi a scuola sono stati semplicemente fantastici. Studiare è stato persino divertente accanto ad alcuni di voi. Certi pomeriggi sono sembrati semplicemente un eterno momento di svago. La cosa più bella è vedere che, anche se gli anni passano, con alcuni il tempo è come se si fosse fermato.

Ringrazio gli amici della Caritas, perché ognuno a suo modo sa insegnarmi qualcosa. La fortuna che ci viene data, di affrontare un percorso così importante tutti insieme, non è cosa da poco. Grazie per la dimostrazione di Fede. Grazie per la dimostrazione di amicizia che va anche al di là del luogo in cui ci siamo conosciuti e che ci vede insieme.

Ringrazio i miei amici dell'università, tutte persone in gamba che mi hanno accompagnato in questo percorso di studio. Il periodo di Erasmus ha rappresentato per me un'esperienza nuova che intimoriva, ma che si è rivelata fondamentale e bellissima allo stesso tempo. Ringrazio chi ha condiviso con me quel periodo, in cui ci si è dovuti supportare e magari, a volte, anche sopportare, in cui si è stretto un legame di amicizia e ci si è imparati a conoscere.

Ringrazio il Prof. Mario Vento, per la sua continua dimostrazione di passione in quello che fa e di dedizione al lavoro. Per i discorsi con cui si impegna sempre a

motivare e spronare i propri studenti, con l'unico obiettivo di farli rendere al massimo. Grazie per la continua applicazione nel trovare soluzioni che permettano agli studenti di fare nuove esperienze formative come, ad esempio, la possibilità di poter usufruire di importanti contatti Erasmus.

A tal proposito, ringrazio il Prof. Tabbone, che nell'arco di tempo trascorso a Nancy si è rivelata una persona bellissima. Grazie per la disponibilità e la gentilezza con cui ha permesso che potessi integrarmi al più presto in un posto nuovo sotto tutti i punti di vista. Per la passione con cui ha seguito studenti che neanche fossero suoi, accompagnandoli nel miglior modo possibile in quella nuova avventura di vita e di studio.

Ringrazio il mio correlatore Antonio Greco, che sia nel periodo di studio all'estero che in quello svoltosi all'università di Salerno, ha sempre dimostrato una grande disponibilità e preparazione. Trovare una persona così non è sempre facile in alcuni ambienti universitari. Grazie per l'umanità dimostrata e, soprattutto, per la tranquillità trasmessa anche quando qualcosa non andava come previsto. Una persona preparata, capace di seguire uno studente insegnandogli quello che sa, come farebbe un amico.

Avrei voluto dire grazie di persona anche a qualcun altro, al mio cane. Purtroppo, o per fortuna, ci si abitua al tempo che passa inevitabilmente lasciando tutto dietro, ma sedici anni insieme non saranno mai dimenticati.

In ultimo, ma non perché meno importante, anzi perché Lui sa cosa spetta agli ultimi, ringrazio Dio per la fortuna di una vita felice ricca di Amore. Per la possibilità che mi ha donato di poter vivere ogni singolo giorno come un giorno felice, un privilegio che viene dato troppe volte per scontato. Mi ha donato la possibilità di rendermene conto e quindi di goderne a pieni polmoni. Da quella prima volta ad Assisi, non è stata più la stessa cosa.

Grazie a tutti coloro che condividono con me questa ed altre gioie.

7 Bibliographical references

1. **Satarupa Mukherjee, BaidyaNathSaha, Iqbal Jamal and Richard Leclerc, Nilanjan Ray.** *A novel framework for automatic passenger counting.*

2. **Thou-Ho (Chao-Ho) Chen, Tsong-Yi Chen and Zhi-Xian Chen.** *An Intelligent People-Flow Counting Method for Passing Through a Gate.*

3. **Chao-Ho Chen, Tsong-Yi Chen, Da-Jinn Wang and Tsang-Jie Chen.** *A Cost-Effective People-Counter for a Crowd of Moving People Based on Two-Stage on Two-Stage Segmentation.*

4. **Bozzoli, Cinque, Sangineto.** *A Statistical Method for People Counting in Crowded Environments.*

5. **Javier Barandiaran, Berta Murguia and Fernando Boto.** *Real-Time People Counting Using Multiple Lines.*

6. **Antic´ B., Letic´ D., C´ulibrk D. and Crnojevic V.** *K-Means based segmentation dor real-time zenithal people counting.*

7. **Senem Velipasalar, Ying-Li Tian, Arun Hampapur.** *Automatic counting of interacting people by using a single uncalibrated camera.*

8. **M. Rossi, A. Bozzoli.** *Tracking and counting moving people.* 1994.

9. **Silva de Almeida S., Cunha de Melo V.** *An Evaluation of two People Counting Systems with Zenithal Camera.*

10. **Juan Serrano-Cuerda, José Carlos Castillo, Antonio Fernández-Caballero.** *Indoor Overhead Video Camera for Efficient People Counting.* Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos, Instituto de Investigación en Informática de Albacete, 02071-Albacete, Spain : s.n., 2013.

11. **Chunhui Tang, Qijun Chen.** *Zenithal People Counting Using Histogram of Oriented Gradients.* 2012.

12. **Jung-Ming Wang, Sei-Wang Chen, Shen Cherng, Chiou-Shann Fuh.** *People counting using fisheye camera.* 2007.

13. **Jung-Ming Wang, Li-Kai Lee, Yun-Chung Chung and Sei-Wang Chen.** *People Counting Based on Top-View Video Sequence.* 2005.

14. **Antonio Albiol, Alberto Albiol, Julia Silla.** *Statistical video analysis for crowds counting.* 2009.

15. **Jaijing K., Kaewtrakulpong P., Siddhichai S.** *Object Detection and Modeling Algorithm for Automatic Visual People Counting System.* 2009.

16. **Gardel Vicente A., Bravo Muñoz I., Jiménez Molina P., Lázaro Galilea J.L.** *Embedded Vision Modules for Tracking and Counting People.* 2009.

17. **Shengsheng Yu, Xiaoping Chen, Weiping Sun, Deping Xie.** *A Robust Method for Detecting and Counting People.* 2008.

18. **Mario Vento, Luca Del Pizzo, Pasquale Foggia, Antonio Greco, Gennaro Percannella.** *A versatile and effective method for counting people on either RGB or depth overhead cameras.* 2015.

19. **Vera P., Zenteno D., Salas J.** *Counting Pedestrians in Bidirectional Scenarios Using Zenithal Depth Images.* 2013.

20. **Dalal, N., Triggs, B.** *Histograms of Oriented Gradients for Human Detection.* 2005.

21. **Huiyuan Fu, Huadong Ma, Hongtian Xiao.** *Scene-adaptive accurate and fast vertical crowd counting via joint using depth and color information.* 2013.

22. **Daichi Kouno, Kazutaka Shimada and Tsutomu Endo.** *Person Identification Using Top-view Image with Depth Information.*

23. **Xucong Zhang, Junjie Yan, Shikun Feng, Zhen Lei, Dong Yi, Stan Z. Li.** *Water Filling: Unsupervised People Counting via Vertical Kinect Sensor.* 2012.