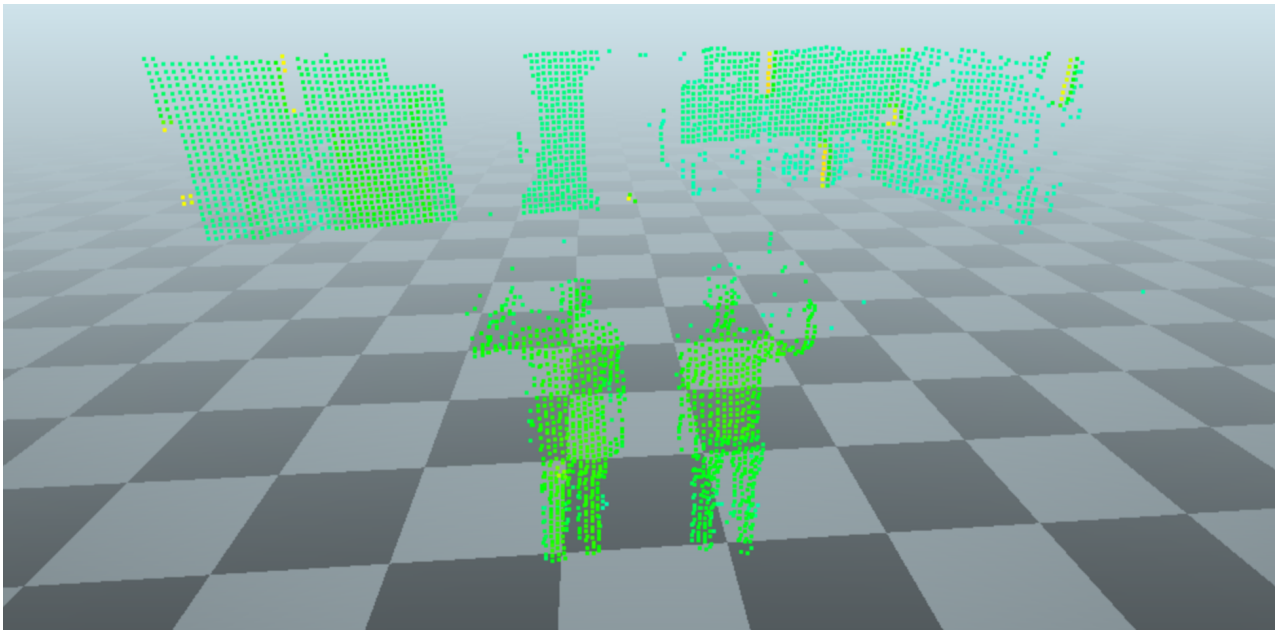
CHALMERS



Clustering and geometrical features for classification of humans in three dimensional data

Master's thesis in Biomedical engineering

DAVID HEDIN
JOHAN WENDEL

Department of Mechanics and Maritime Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2019

MASTER'S THESIS IN BIOMEDICAL ENGINEERING

Clustering and geometrical features for classification of humans in three dimensional data

DAVID HEDIN
JOHAN WENDEL

Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2019

Clustering and geometrical features for classification of humans in three dimensional data
DAVID HEDIN
JOHAN WENDEL

Master's thesis 2019:66
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:
Picture of the authors captured with the time of flight depth camera used for the project

Chalmers Reproservice
Göteborg, Sweden 2019

Clustering and geometrical features for classification of humans in three dimensional data
Master's thesis in Biomedical engineering
DAVID HEDIN
JOHAN WENDEL
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology

Abstract

Computer vision is a rich research field that uses images from normal or technically specific cameras to perform tasks ranging from surveillance to autonomous driving. Computer vision with depth images are however relatively new. Depth images add a third dimension to the image by giving every pixel a depth value and can be produced with several different camera types. The implications for depth imagery is that if an object can be classified in a depth image as e.g a trashcan, a cat or a human, the nature of the data immediately also gives us the distance and position of that object. The extra data from these cameras can enable estimating the volumes and sizes of objects in an image with some extra processing. The focus of this master thesis is on the processing and analysis of this depth data to enable object identification and human classification.

Much research has been on analysing 3D data from the 2D perspective, in this thesis the captured data is first converted to Cartesian coordinates before attempting classification, yielding further possibilities.

The goal of this thesis was to find if there are some anthropomorphic-geometrical features that can describe the human body well enough to accurately classify humans in the Cartesian data. The features are used in two ways, as a Heuristical-geometrical filter and as features for a support vector machine.

Furthermore the thesis presents a successful dynamic adaption of the fast-DBSCAN (Density Based Spatial Clustering of Applications with Noise clustering) algorithm for 3D Cartesian data and a slice method for finding local maxima of point cloud objects.

The results show that anthropomorphic-geometrical features can to an extent be used to classify Cartesian point cloud data. Low resolution cameras has potential for classification purposes as resolution seem to have little effect on geometrical classification as long as human resolution is no less then 20px vertically. Some further work would be needed to create a anthropomorphic-geometrical for real world application.

Keywords: Depth camera, moving platform, human machine interaction, localisation, clustering, human classification, 3D,

Acknowledgements

We want to thank Albin Pålsson our supervisor at Kollmorgen and Pinar Boyraz Baykas our supervisor and examiner at Chalmers for helping and guiding us with their expertise. Kollmorgen for giving us the opportunity to do this project and using their facilites and Chalmers University for preparing us for it. We also want to thank each other for always supporting each other to press forward and continue working.

Abbreviations

AGV : Automated Guided Vehicle

DP : Data Points, points in the point clouds containing data such as coordinates and index.

FN : False Negative, an object supposed to be identified but was not identified.

FP : False Positive, an object supposed not to be identified but was identified.

HMI : Human-machine interaction, term for how humans and machines communicate their intentions between each other

ROI : Region of interest, used for referencing a sub region of a larger image of especial interest for an application.

SVM : Support Vector Machine, a type of supervised machine learning

ToF(camera) : Time of Flight, the type of camera used in this project

TN : True Negative, an object supposed not to be identified and was not identified.

TP : True Positive. an object supposed to be identified and was identified.

Contents

1 Introduction

1.1 Purpose

The aim of this thesis is to construct a system for detection and localisation of humans in point cloud data, using especially low-resolution point clouds derived from depth images. The end-goal of the project is to improve the AGV motion planning and behaviour algorithms, incorporating the human-detection and localisation system as a sub-module for increasing safety in human-AGV interaction. In addition to the algorithm development based on low-resolution point-cloud data, an in-depth literature study will be performed to gather state-of-the-art methodologies in design of Human-AGV interaction. We will review (in a systematic manner) the early studies of such interaction and communication interfaces to identify the best practices to be utilised by an AGV working in a modern warehouse.

1.2 Demarcations

The image pre-processing algorithms used in this project are partially developed by the project team. Some of the data pre-processing is provided by the internal software of the Time of Flight(ToF) camera used in the project. The algorithms utilised by the ToF camera are briefly described in Section 3.1, but not covered in full in this thesis report for the sake of brevity. Original algorithms developed by the project team is emphasised throughout the thesis manuscript where it is appropriate. It is assumed in the proposed algorithms that the data acquisition device is installed at a known distance from the ground. Furthermore all testing was performed in controlled environments and a indoors hall illuminated by both lamps and natural light from windows as it best replicates real use environments.

1.3 Short summary on the subject of depth images

Giving images a sense of depth goes back to the two-lensed stereoscopes which were used as entertainment during the 1800s. The concept is not dissimilar from how a modern 3D movie is filmed. The present day range or depth camera as it will henceforth be referred to, adds a distance to images. The result is a grey scale image where pixel intensity represents a depth value. Two different techniques to achieve this are: Stereoscopic vision utilising two lenses at a known distance apart, now so common that they appear in many smart phones. The second technique type is Time-of-Flight(ToF) where distance is measured by sending out near infrared light. The ToF technique was used for filming the training data used in this project and is explained in-depth in section 2.2.

1.4 Earlier work

Analysis on 2D images has been around at least since the 1960s, but general interest for using depth cameras for image analysis spiked with the release of the Microsoft Kinect that made depth cameras affordable to the general public. Kinect cameras has a relatively high resolution but would be unfit for safety implementations because of the short range. The camera has been used in in several research publications for the purpose of human detection and motion tracking. M.H.Khan et al [18] proposed matching a template with the edges of a human head onto an image processed with Canny edge detector[4]. The points representing possible humans in the image are narrowed down by applying a geometric filter shaped as a simple hemisphere, capable of sorting out flat-surfaces as proposed by L. Xia et al [30]. As is shown the algorithm performs well on their data sets where all humans appear close to the camera. One can assume it lacks scaleability and classification would perform worse with this approach for lower resolution images. The reason for this is that the relative difference between the templates becomes too small and calculating correlation between image and template is no longer an effective tool at lower resolutions. The geometrical comparisons as proposed by Wang[30] are however not as dependent on resolution as they do not use correlation as measurement.

An alternative or complimentary approach to depth images are point clouds for which the theory is explained in depth in section 2.3. In Z.Yan et al[31] data is collected with a lidar scanner and humans are distinguished

using a clustering algorithm to separate the point cloud into clusters. For classification or distinguishing the humans from other objects in the scene, the machine learning technique SVM(support vector machines) is used. Six different object features are used in training. This is combined with a tracking algorithm. In the paper Z.Yan et al[31] make effective use of the cartesian distance by making dynamic calculations that change depending on object distance from the sensor. Result shows an effective approach, however it is important to notice that the SVM depends heavily on what features are used.

A different machine learning method is Deep Neural Networks(DNN) which is the most successful algorithm for image classification in current research field. Though the method existed earlier DNN got widespread recognition with AlexNet [20]. DNN requires un-supervised training for a long time on on very large datasets with labels. With a limited amount of labelled point cloud datasets, in particular for point clouds derived from depth cameras, the DNN approach is not feasible within the given time limit of the project.

2 Theory

2.1 General camera theory

The most fundamental concept in a camera is the one it gets its name from, 'camera obscura' meaning dark chamber. It has been known since antique times and the concept was first recorded in the Chinese Mozi scrolls from 400BC. If a small hole is drilled into the wall of a dark chamber it will project an image of the lit outside world onto the opposite wall inside of the dark chamber. The concept is shown in figure 2.1. In a modern camera, the chamber is the camera housing and the hole has been replaced by a lens. The wall that light was projected on is now either photosensitive film, or in the case of digital cameras the image capturing function is provided by electrical photosensitive image sensors. The number of image sensors directly correspond to image resolution, e.g 100 rows and 100 columns gives a $100x100 = 10000$ pixel resolution image.

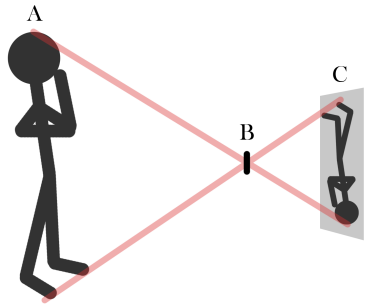


Figure 2.1: The figure illustrates how light reflects from subject(A), goes through the pinhole(B) and hits the sensor(C).

2.2 Distance measurement and imaging with time of flight method

Time of Flight(ToF), is a method used to measure distance. The word has several different applications that could refer to either light, sound or ballistic objects. In this section ToF will refer to measurements using light.

The camera sensor setup can be combined with ToF measurement technique to make a ToF-camera that captures 'range' or 'depth' images. Each pixel in this depth image has an intensity value corresponding to the distance. The image looks like a heat map representing the distance from the object to the camera. An example of this can be seen on the left image A in figure 2.4.

There are several and slightly different techniques to make the ToF measurements but all include active illumination (sending out a light burst), one or more light sensitive sensors and a timer capable of measuring very small time windows.

The first type of ToF type is 'Direct Time of Flight' also called snapshot or trigger mode. This type of ToF records the time between sending a light burst and receiving the reflection. Since the speed of light is known calculating the distance can be done as shown in equation 2.1, where c represents the speed of light in air and t the measured time window. Using the direct time of flight method the whole three dimensional scene is recorded with a single pulse from the camera illumination source.

$$D = c \cdot t \tag{2.1}$$

Second type of ToF is 'Range Gated ToF'. They use shutters or 'gates' that opens and closes as the light pulse is sent. When the gate closes some of the returning light will be blocked and the difference between how much light was received and blocked can be used to calculate distance. Equation 2.2 gives the distance, where R is camera range S_1 is amount of light received and S_2 is light blocked by the gate.

$$z = \frac{R \cdot (S_2 - S_1)}{2 \cdot (S_1 + S_2)} + \frac{R}{2} \tag{2.2}$$

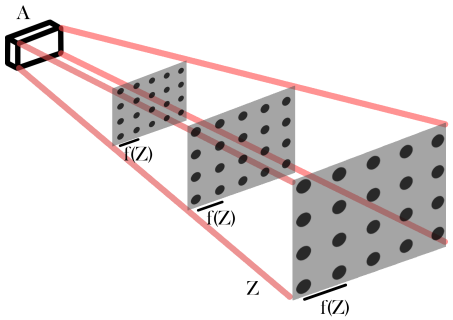


Figure 2.3: Illustrating how area coverage increase but also pixel relative distance as object move further from camera

The active illumination source in ToF measurements can be either laser or LED and use light in frequencies in or near the infrared spectrum. This decreases the possible of interference with the sensor from other light sources and reflections, since not all light sources can produce light in infra red frequencies. This can be combined with a lens containing an optical band-pass filter, restricting frequencies received to those used by ToF.

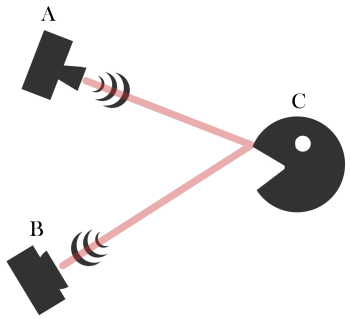


Figure 2.2: Figure illustrating how active illumination source(A) fires at subject(C) and light reflects and hits sensor (B). By measuring the time from A firing to B receiving light signal it is possible to determine the distance the laser has travelled.

To create a depth image, several ToF sensors are combined into a grid, like in the digital camera setup explained in section 2.1. Depth images has a unique property that can be used for image analysis. Compared to a regular black and white or RGB(red/blue/green) image, all objects will have well-defined and mathematically separable edges relative to the background regardless of colour and scene illumination. For example in an RGB image a human might have a shirt with similar colour to the background making the pixel value difference at the edge small. However, in a depth image of the same scene this difference would be significant as the pixel value in depth image is created by distance instead of colour.

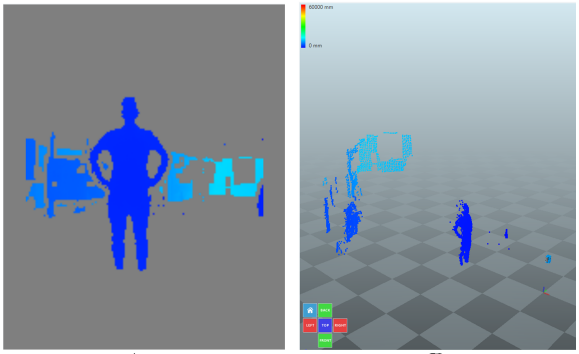
2.3 Description of Point clouds

Three dimensional data acquired by a range camera(section 2.2) or a LIDAR(Light Detection and Ranging) laser scanner(similar to range camera but with 360 view) can be visualised in a three dimensional space. The standard way of representing this is a point cloud, where each pixel in the depth image is represented as an Cartesian (x,y,z)-point in space with origin centred on the camera sensor plane. An image taken by a range camera is converted to these Cartesian coordinates using the ToF-camera parameters. With knowledge of the amount of horizontal and vertical pixels as well as camera view angle, the image pixels can be converted to Cartesian coordinate format in a point cloud.

Point clouds create new possibilities for working with the range data. The data can be analysed in three dimensions and the real world geometry can be used for analysis and classification. It also presents the possibility of analysing the data in two dimensions but from a different direction, e.g instead of viewing from x,y with z as pixel intensity(depth) it can be viewed as z,y with x as pixel intensity, making it appear as the scene is viewed from the left. This is however not utilised for this paper.

Figure 2.4 shows an example of a depth image converted into a point cloud with Cartesian coordinates. In both a depth image and a normal image the image is made up of equally sized pixels. When an object is

further away from the camera it appears smaller. Our own eyes function in a similar way but we know that objects do not actually get smaller. When a depth image is converted to a Cartesian coordinated point cloud all the pixels are rearranged to their real world positions. This causes the small objects from the image to be scaled up to their real size. From the front the distance of object from the camera is instead visually shown by their point density as shown by figure 2.5 and 2.6.



A B

Figure 2.4: Figure showing A: Original 2D depth picture and B: Point cloud made from the picture and shown in a 3D space.

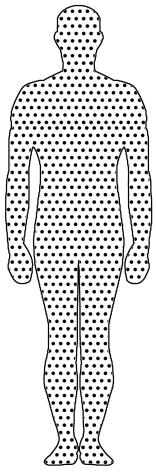


Figure 2.5: Example of point density on a human close to camera

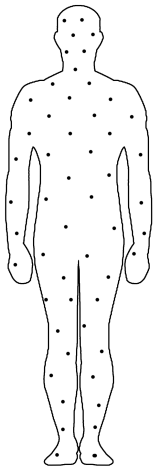


Figure 2.6: Example of point density on a human far from camera

2.4 Support vector machines

This section will provide a quick introduction to support vector machines(SVMs). First introduced in a state similar what is used today by Bosher et al in 1992 [2].

SVM is a supervised machine learning algorithm. The goal is training a classifier to automatically classify data input into different categories. Supervised machine learning means that the classification algorithm is trained and tested on a set of pre-labelled data. This data set is called a ground truth. Because of this labelled ground truth data set it is simple to compare if the classified category output from the algorithm is correct.

As with any other machine learning method the object of SVM is to correctly classify the input data points into different categories. Each data point or 'observation' is represented by different variables or 'features'.

These features can be anything: coordinates, colour, age etc, it all depends on what kind of observations are being analysed. Together these features form an n-dimensional space that ideally makes it possible to separate the observations perfectly into categories. The features heavily limits the theoretical best performance of the SVM. Tuning the SVM parameters or changing amount of training data can not compensate features that can not adequately describe observations enough to differentiate them.

In short, the SVM tries to create a hyperplane, that separates the n-dimensional feature space into several categories. This limit is easiest to imagine in two dimensions where the separating limit of the data points would be a line or in three dimensions as a classic plane. For higher dimensions we view it as a more abstract mathematical plane.

The 'Support vectors' that has given the algorithm its name are the data points closest to the dividing hyperplane. They are the points most difficult to classify and removing just one of them would change how the hyperplane is drawn. The hyperplane is calculated as an optimisation problem that maximise the margin between the hyperplane and these support vectors ideally placing it exactly on the border that defines the difference between the classes. In figure 2.7 an illustrated example of an SVM is shown, the observations being represented by two features resulting in a one dimensional line separating a 2D space. Soft margins were introduced by Cortes et al [6]. In summary they allow the calculation of a hyperplane that is allowed to miss-classify a few observations in an attempt to avoid overfitting the hyperplane to the training data.

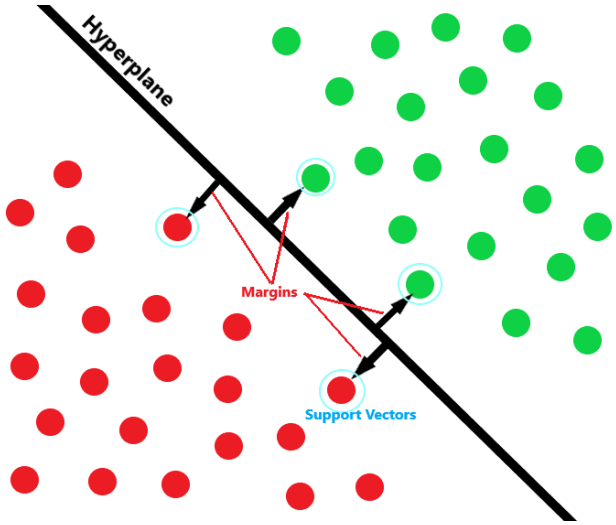


Figure 2.7: A two dimensional visualisation of the hyper plane binary classifying data.

2.5 Litterature study on human-machine interaction and intent communication

In this part we will present a literature study on the findings of previous works on humans behaviour and actions regarding vehicles, signals, messages and other people. This section will form a base for a discussion the last chapter regarding how warehouse AGVs warning/information systems and behaviour should be designed. The study was requested by the case company and is intended to gather information as support for how a more holistic AGV human awareness-system could be designed taking more than just the sensors into account.

As there exist little to no research on how AGVs interact with humans the majority of the literature used is about how automatic cars interact with humans. There are many parallels between automatic cars and AGVs and within reason it can be assumed that humans react and behave quite similarly between the two even though most AGVs and cars have big a size difference and average velocity.

2.5.1 Current behaviour of AGVs

Behaviour for AGVs differ between manufacturers and such there is no description that covers all the variations but a lot models share some similarities. Today they need to fulfill some minimum requirements [3] [13].

- Stopping if an object is about to enter the stop zone, if the object suddenly appears in the stop zone the AGV is not required to do anything.
- To detect obstacles it need to use either a bumper that will not exert more than 134N or a non-contact sensor device that makes sure the AGV does not crash into something or one of these options paired with other devices.
- An AGV should leave 450mm of free space between itself and obstructions.
- The AGV is required to do an emergency stop if it goes too far off the intended path.
- During certain conditions the AGV need to create a clear warning that can be audible, visual or both. Conditions include when the AGV starts, moves, changes direction and for control system failures.

Today common practice is using measuring lasers to avoid obstacles. Successful simple trials have been done using a 3D camera to detect obstacles [14], in the future more advanced trials might be conducted. A danger of only using measuring lasers is that they are often installed at the bottom of the AGV and thus can not detect objects in the air. Objects in the air might have footing on the ground outside the lasers range while the object is still in the way for the AGV. Examples include a wide table with four legs on the corners where the AGV would not detect the legs but collide with the tabletop and the forks of a forklift where the vehicle is outside the lasers range but the fork with its load is in the air in front of the AGV and will cause collision.

2.5.2 Cars informing humans on their intent

It has been shown in research of cars interacting with humans that humans prefer to act according to what they have learned previously in traffic safety even if there are new methods such as signs on the cars showing if it is safe to cross the street[5]. Signs used in traffic are often regulated by law and have a specific design which needs to be easily recognised by humans. When the humans in a defined environment (such as traffic or warehouse) see a new sign, confusion can occur and thus could be a contributing factor to people not acting on it[5]. If there was a standard for signs on vehicles it would be more effective since observers could immediately recognise it and what it means in the same sense traffic signs are recognised today. People crossing the street seem to prefer using eye contact with the driver or difference in speed of the vehicle in order to make a decision to cross or not [25]. When in a group people, pay less attention to their environment and about traffic rules. If a group of people of three or more cross the street while the traffic light is red other people waiting will also follow suit and cross and therefore also break the rules.

Tests have been made on signalling from cars to find out what humans can understand and prefer [27]. The results showed that text was the best to not have any misunderstandings but only if worded in a very instructive way, otherwise it was detrimental. However, in other tests text and symbols was equivalently informative when it come to understanding it intuitively, reaction time and certainty of content and it was recommended that a standard for symbols used in traffic should be found as to improve reaction time and certainty of content.

It was tested how to display the signal on different parts of the car and project it on the street close to the human. From these tests the projection got the best results but it was noted in the report that it might be unpractical since bright light like sunshine can make the projection harder to see.

In the same paper it was also tested how behaviour like acceleration and deceleration can be used as signals. The tests showed that it is difficult to notice when a car starts accelerating but can quite reliably notice when a car decelerates. The people in the test reacted the quickest when the car slowed down and they also could notice the sounds from the car as it changed speed. The participants also found it unsettling when the car drove against them in a slow but constant speed.

2.5.3 Sounds made by electric cars

European commission guidelines

The United Nations Economic Commission for Europe (UNECE) have adopted guidelines for how Acoustic Vehicle Alerting Systems (AVAS)[11] mounted on cars should sound. Mainly it is one sound for all speeds over 20km/h, one sound for for speeds between 0 to 20km/h and one sound for reversing. These sound profiles are not allowed to be mistaken for any kind of emergencies as a siren or emergency vehicle or be an intermittent sound and should avoid sounds that can be confusing like animal sounds or a sound that gives the impression of deceleration when the car is accelerating.

How electric cars should sound

A test performed for 17 different sounds an electric car can make to see which one was the most appropriate and effective with 40 participants. The conclusion was that an electric car should sound like a combustion engine, or "like a regular car" as a sound people are already familiar with. While seeing what was the most preferred specific sound there was no clear conclusion as preferences varied widely between the participants. They also wanted as follow-up research on how to design the engine sound and do tests if it is possible to adjust and exaggerate characteristics of the sound to improve reaction times to hearing it[23].

2.5.4 The effect colour has on humans

A study regarding emergency vehicles researched how a human can detect different colours and what they are associated with socially [1]. It showed that white, yellow/amber, red and blue in that order is the brightest and easiest to detect. They came to the conclusion that alternating between red and blue light is the optimal in both low and normal light conditions. Another conclusion was that the colour of the light affected how the person seeing it reacted and it was closely linked to what they have learned from experiences. In states where police used red lights people reacted strongly to all red lights and in states where police used blue lights they instead reacted to blue lights but not to red.

There has been research on how different colours affect humans psychologically but while there seems to exist correlation there is so far nothing conclusive to show [9].

3 Method

This chapter will propose two different human classifiers that builds upon the same geometrical human features. Both rely on the same clustering algorithm first splitting the recorded data point cloud into separate objects. Furthermore the chapter will present the hardware used when recording and processing the data as well as present the data set used for training and testing the classifiers.

3.1 Camera specifications

The camera used for this project is a 3D vision Visionary-T V3S100-2AABAAB[28] made by the company SICK. Its primary uses in industry is not image classification but making topography checks of a surface, used for checking how packed a container is or if product surface has been cut/sanded correctly. Commercially there are not many 3D cameras promising a long range usability and SICK cameras are made for industrial use. Compare this to the popular 3D camera Microsoft Kinect that has a relatively high resolution but a maximum range of 7m and is built to be used in the home. Part of this project will be to evaluate this cameras suitability for classification purposes. For the rest of the paper the Visionary-T V3S100-2AABAAB will simply be referred to as 'the camera'. The following list includes the cameras parameters that will be discussed later or used in calculations. Setup can be seen in figure 3.1. In the camera software some image processing functions are included. These are aimed at decreasing noise like removing lonely floating pixels. The project has only made use of the region of interest 'ROI' filters explained in 3.5.1.

Working distance	0.5m to 60m
Detection angle	69°horizontal x 56°vertical
Angular Resolution	0.39°horizontal x 0.38°vertical
Light Source	Invisible infrared light (LED, 850 nm)
Pixel count	176 px horizontal x 144 px vertical
Dimensions (L x W x H)	162 mm x 93 mm x 78 mm
Light sensitivity	$<$ 50 klx, Sunlight

Table 3.1: Camera specifications from product's data sheet

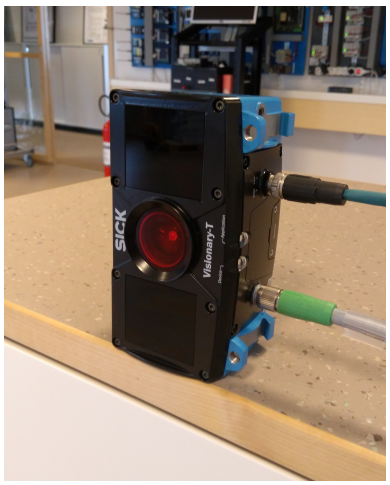


Figure 3.1: The camera was placed standing on its side for recording since cables go into it from beneath and no mounting was used

3.2 Processing unit used in testing

The data processing was performed both online and offline on recorded datasets. Computations software was Matlab 2018b using a Dell laptop with:

- Processor: Intel(R) Core(TM) i7-8650U CPU @ 1.9Ghz 2.11GHz
- Usable RAM: 15.9 GB
- Operating system: Windows 64-bit

3.3 Depth-datasets used in testing

For training and testing the classification algorithm several videos where recorded and labelled. The recorded data is split into two sets, one for training with 1650 frames and one for testing with 415 frames.

3.3.1 Our ground truth dataset

The current standard for AGVs require floor height mounted sensors to avoid collisions. The safety test for vehicles equipped with these intend to make sure that the vehicle can not harm a human. For testing this, the AGV has to detect and stop for static object with dimensions chosen to represent a standing leg and a human laying down, shown left side in figure 3.2. These tests are designed for simple lasers but a depth camera is capable of viewing more of the scene and give a more detailed picture. To enable more advanced behaviour humans should be distinguishable from surrounding areas and be detectable in several common positions.

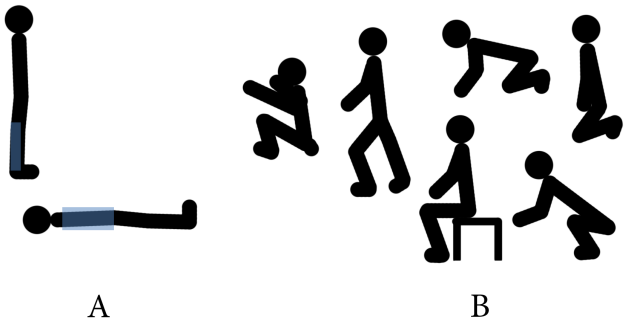


Figure 3.2: The blue in A is the old tests and b is poses for new tests that are crouching, walking,kneeling down, kneeling up, sitting and squatting.

The recorded and labelled data set contains humans in several true-to-reality workplace situations that the AGV could encounter. They include humans moving around in an open space, picking up objects from the floor and carrying them, carrying large objects(ladder), humans moving behind other humans or objects and humans sitting down.

3.4 Details on evaluation process of the proposed algorithm

To test the algorithm the sensitivity, specificity, accuracy and computation time were evaluated. How these parameters are measured are presented further down in this section section.

New features were added to the algorithm in an iterative way. If a rough version of the feature showed initial promise, they where kept and subjected to parameter tuning to see if it could be improved. As a rule of thumb an early promising feature was when testing on recorded data the added feature removed a significantly larger percentage off false positives(FP) than true positives(TP), approximately four times the number of FP removed than TP.

3.4.1 System computation time

Testing of algorithm computation time was performed in Matlab with its built in time function. The test ran on the pre-recorded test data set and divided by number of frames in the video to get mean processing

time. Only algorithms combining features that performed faster than 500 milliseconds per frame where kept for further evaluation others where discarded. 500 milliseconds was chosen as the limit based on a rough estimation of how quickly a slow moving AGV need to respond in a real world scenario to give a reasonable braking distance.

3.4.2 Checking 3D classifier performance against 2D labelled ground truth

For repeatable offline testing, prerecorded videos and their frames are used as input to the system. The camera captures a depth image that is then converted to a Cartesian point cloud, therefore all point cloud frames has a depth image equivalent. The ground truth is manually labelled in the images using MATLAB software. As all labelling had to be done manually by the research team, it was chosen to do it on depth images instead of in the Cartesian point clouds. This was to save time as there was no available tools for labelling in 3D so labelling in 2D images was more feasible.

Classification uses the point cloud, the algorithm calculates and outputs an estimated centre of the human object in cartesian coordinates. This is converted to a pixel in the labelled ground truth depth image, see figure 3.3. If the pixel lies inside the rectangle outlining a human and is the only point inside the rectangle it is regarded as correct, if it is outside a marking it gets a false positive and for every extra marking inside a rectangle a false positive is added. The datasets and what scenarios they include are more thoroughly explained in section 3.3

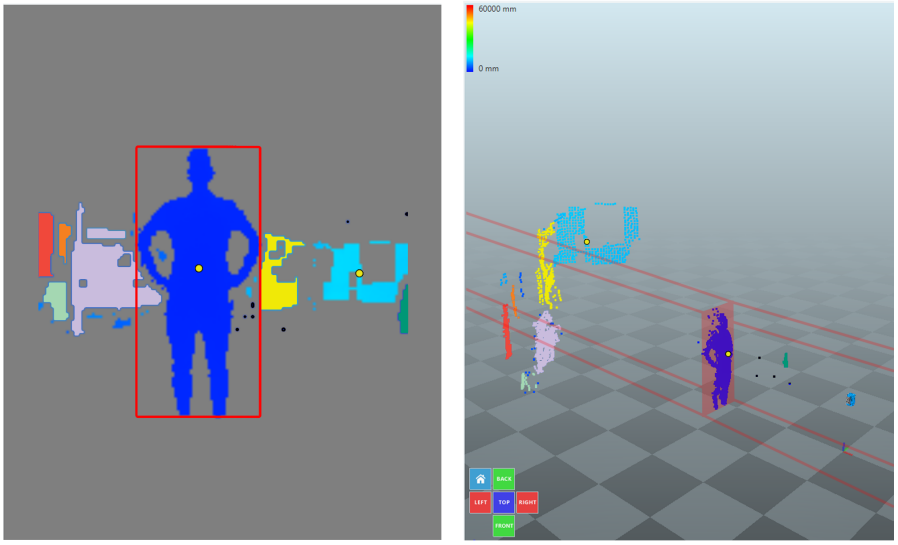


Figure 3.3: Figure illustrating the marked human ground truth in the 2D image and yellow dots in the 3D point cloud representing human guesses. The dot on the human is within the ground truth box so it would be a TP and the dot on the teal wall is not in a box and would be a FP.

3.4.3 The evaluation parameters: precision, accuracy and recall

The most basic way of evaluating the performance of a classifier is looking at

TP	True Positives:	correctly classified as human
FP	False Positives:	falsely classified as human
TN	True Negatives:	correctly classified as non-human
FN	False Negatives:	falsely classified as non-human

If used as individual number they say very little about a system, however when combined they give precision, shown in equation 3.1, accuracy shown in equation 3.3 and recall shown in equation 3.2. Together these give a better view of the system performance. These measurements are used when presenting the results from the two classifiers in section 4.2.

$$Precision = \frac{\text{True Positives}}{\text{True Positives + False Positives}} = \frac{\text{True Positives}}{\text{All classified as positive}} \tag{3.1}$$

$$Recall = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} = \frac{\text{True Positives}}{\text{All Actual Positives}} \tag{3.2}$$

$$\begin{aligned} BalancedAccuracy &= (\frac{\text{True Positives}}{\text{True Positives + False Negatives}} + \frac{\text{True Negatives}}{\text{True Negatives + False Positives}})/2 \\ &= (\frac{\text{True Positives}}{\text{All Actual Positives}} + \frac{\text{True Negatives}}{\text{All Actual Negatives}})/2 \end{aligned} \tag{3.3}$$

3.5 The human classifier in detail

The following section will explain all the parts of the human classification algorithm in detail. Two different human classifiers will be presented that build upon the same geometrical human features. One is called the Heuristic-Geometrical approach(section 3.5.4) and the other the Support vector machine(SVM) approach(section 3.5.5). An overview of the complete system can be seen in figure A.1 in the appendix, the different parts of the flowchart are each explained in their own subsection. Both classifiers rely on the same pre-processing(section 3.5.1) and a clustering algorithm that first split the recorded data point cloud into separate objects(section 3.5.2). After classification two additional filters are applied and a tracking algorithm is applied to each classified human (section 3.5.6)

3.5.1 Pre-processing for efficiency increase

For a safety application for an AGV operating in real time, processing time for a recorded frame until making a decision can not exceed a certain time limit. This time limit depends heavily on factors such as speed. In this project, 500ms was used as a soft upper limit based on a rough estimation of how quickly a slow moving AGV need to respond in a real world scenario. This creates a need for computational solutions that decrease computation time. To achieve this, several pre-processing algorithms was considered to decrease the amount of obtained data points of no or little interest i.e having no chance of being part of a human object or not essential to classifying one.

Defining a region of interest(ROI)

Using a point cloud in in Cartesian coordinate format, it is possible to know where each data point is in space. By introducing a 3D box in space sorting out the objects of interest(OOI), many points can quickly be discarded to decrease the demand on future computations on the data. This 3D box is called our region of interest or ROI.

The bottom edge of the ROI is a limit that discards everything from 10cm and below. This will remove the floor. In addition to decreasing computation time, removing the floor is essential as it removes the connection between objects in the 3D space and this makes it possible to cluster them.

Any point found closer than 1000mm in front of the camera is removed. This is because early tests of the camera (presented in the result chapter) revealed a high presence of noise pixels close to the camera

Thirdly a limit is set on the assumption that no human being will be higher than 250cm. Removing data points above 250 also results in removing the roof or upper parts of high walls which can make up a significant part of the data points.

Randomly dropping points in the point cloud

To avoid unnecessary strain on the clustering(section 3.5.2) and classification algorithms(section 3.5.4 and 3.5.5) it is desirable to have as few points as possible in the point cloud without majorly impacting performance. The proposed solution is dropping points randomly to decrease computation load. Random here

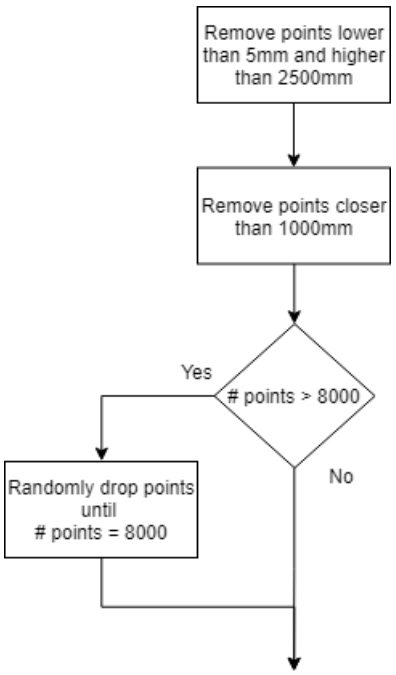


Figure 3.4: Flowchart of the steps involved in the pre-processing

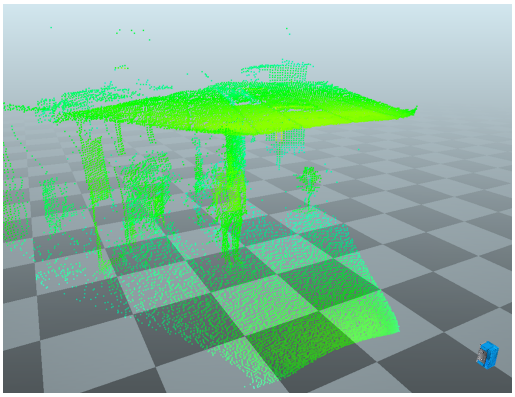


Figure 3.5: Point cloud without ROI filter

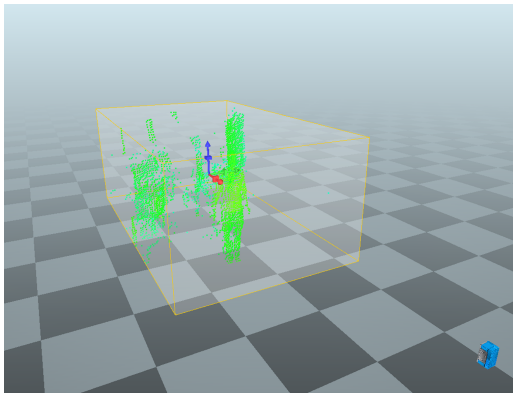


Figure 3.6: Point cloud with ROI filter

means each point has the same chance of being dropped. In the point cloud(explained in section 2.3) objects closer to the camera will have more points than objects farther away. For safety reasons, it can be assumed that the objects close to the camera are of higher importance for collision avoidance algorithms. Most of the points will be removed from close objects since these contain more points. Because of this difference in objects density in the point cloud, the objects will not change much as long as the total amount of points is kept above a reasonable threshold(Results, section 4.1). Therefore, dropping points is an effective way of keeping computation time down.

To evaluate how many points the hardware (section 3.2) could run the algorithm on while keeping time per frame below 500ms, a wall was recorded giving all pixels and system was run while clocked. At 8000points the algorithm performed consistently below 500ms. In section 4.1 the impact on performance from different percentages of points dropped is presented.

3.5.2 Clustering algorithm, separating the point cloud into clusters

Before any type of classification is performed on the data, the point cluster need to be divided into smaller clusters where ideally each one represents a separate object in the data. This can be performed by a clustering algorithm. There are many different existing clustering algorithms, the one we present here is based on the fast-DBSCAN algorithm proposed by K.Mahesh Kumar et al [21]. That in turn builds upon the DBSCAN(Density Based Spatial Clustering of Applications with Noise) created by Martin Ester et al[10].

Following is an explanation of the theory behind the fast-DBSCAN, for a visual clarification see the complete flowchart of how each point is assigned to a cluster in figure A.2 in the appendix.

Step one, all points start as unassigned. The clustering algorithm goes through all unassigned points in the point cloud once and searches in a $3r$ (three times radius r, defined further down) distance around each of them for a master point. If there is not a master point within this distance the current point becomes a new master point and is assigned a masterpoint number, this same number will be assigned to all future slave points to this master point. If there is a master point within $3r$ and it is within r it is assigned to that master point as a slave. Lastly if there is a master point within $3r$ but not within r the current point is left unassigned.

Step two, the clustering algorithm goes through all unassigned points from previous step once. If there is a master point within r distance of the current unassigned point it is redefined as a slave to that point. If now master point could be found within r the unassigned point is made into a new master point.

Step three, go over all master points once. If a master point has another master point within $7r$ horizontal or $5r$ in depth or vertical, the master point and all its slaves are assigned to the other master point adopting the master point number. The end result is several groups of combined points with the same number that mak up the different objects in the point cloud data. Distances $7r$ and $5r$ where chosen through heuristic testing what showed to most correct clustering.

The fast-DBSCAN algorithm explained above was created for two dimensional data and use a constant radius r when searching for neighbouring points. Having radius r as a constant is not suitable when working with data derived from point clouds, because the distance between points increases as they get farther from the camera. This makes objects closer to the camera appear more dense, the concept was illustrated in figure 2.4 in the theory section.

The increasing distance between between adjacent points in the point cloud depending on depth creates a problem for the algorithm. To solve this we propose a dynamic radius r_{dyn}. The equations for this dynamic 'search sphere' as it appears in 3D space can be seen in equations 3.4, 3.5 and 3.6, where α and β are the horizontal and vertical camera angles, d is the current point distance from camera and p_h p_v are the camera

resolution in pixels horizontally and vertically. Figure 3.7 show how this calculation is a Pythagoras problem.

$$r_{width} = \frac{2 \cdot tan(\alpha) \cdot d}{p_{horz}} \tag{3.4}$$

$$r_{height} = \frac{2 \cdot tan(\beta) \cdot d}{p_{vert}} \tag{3.5}$$

$$r_{depth} = r_{width} \tag{3.6}$$

3.5.3 Features used for classification of humans

This section explains the features used for the two different human classifiers that where tested. Both human classifiers builds upon a collection of geometrical features meant to represent human physical characteristic. In the first classifier the features together with a threshold are simply applied on each of the clusters that was the output of the clustering algorithm explained earlier in section 3.5.2. If a cluster passes all the tests the cluster it is classified as a human. The second approach uses support vector machine(SVM) for the classification. SVM is a supervised machine learning algorithm, the general is explained in the theory section 2.4. How the features are applied to the SVM are explained in this section.

For the system to be able to recognise a human, we need to define the conditions/scenarios under which the algorithm can successfully recognise a human in a static or dynamic world scene. These limitations are mainly about how visible a human need to be and can be found in section 5.2 in the discussion chapter.

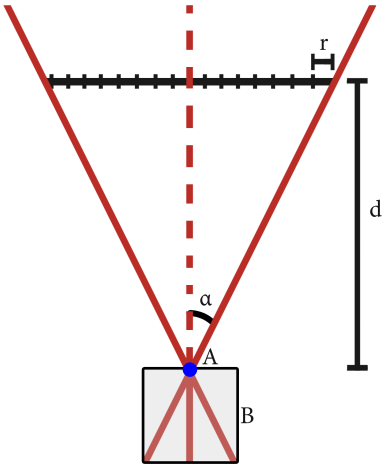


Figure 3.7: illustration of the different variables in equation 3.4 r being the distance covered by one pixel in the image at distance d

Possible head locations

Both classifiers build upon searching the clusters (section 3.5.2) for possible locations of a head, see figure 3.8 for a visual reference. The possible locations of a head are found by searching for local maxima among the y-coordinates of points in the top portion of the cluster. The 3D cluster is sliced vertically and the top points are retrieved from every slice. An algorithm to find the local maximum is applied on the array of top points. Since there is always some noise present, even a flat surface will produce one or more local maxima. These relatively small local maxima however are of no interest so local maxima are further filtered by demanding that each local maxima have a prominence value over a certain threshold. Local maxima that remain are saved as possible candidate locations for a head and are used to retrieve more features explained in the next sections.

Geometrical features

Variable features Variable features are retrieved from the whole cluster, where the cluster is made of n amount of data points each defined in Cartesian coordinates x, y and z. These features are not reliant on the headpoints explained in 3.5.3. All of the variable features can be seen in image 3.10.

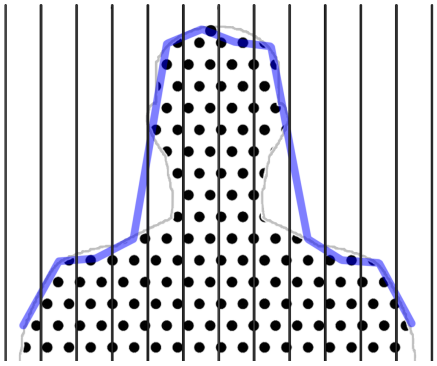


Figure 3.8: Illustrating how the top points are recovered to search for possible head locations resulting in a top curve. Local maxima with a prevalence over a certain threshold are then retrieved as the possible head locations.

- *Centermass*: the mean of all datapoints in X, Y and Z.

$$\hat{x} = \frac{1}{n} \cdot \sum_{k=1}^{n} x_k \tag{3.7}$$

$$\hat{y} = \frac{1}{n} \cdot \sum_{k=1}^{n} y_k \tag{3.8}$$

$$\hat{z} = \frac{1}{n} \cdot \sum_{k=1}^{n} z_k \tag{3.9}$$

- *Max* and *Min* X: the values of the highest and lowest data points seen in the horizontal perspective.

$$max(Cluster(x)) \tag{3.10}$$

$$min(Cluster(x)) \tag{3.11}$$

- *Max* and *Min Z*: the values of the highest and lowest data points seen in the depth perspective.

$$max(Cluster(z)) \tag{3.12}$$

$$min(Cluster(z)) \tag{3.13}$$

- *Min Y*: the lowest value of the data points seen in the vertical perspective.

$$min(Cluster(y)) \tag{3.14}$$

- *Head width* and *Head depth*: width is calculated by making slices downwards ending 15cm below the head point(Est. where the widest part of the head should be). It will continue making slices in both directions until a slice end up empty. Then it will from the horizontal view compare the datapoints to the lowest and highest values and calculate the width by using the difference between the points. The depth is the datapoint within slices with the lowest depth value. Concept is illustrated in figure 3.9

- *Shoulder width* and *Shoulder depth*, same as with *Head width* and *Head depth* but taking all points 60cm below Possible head point instead of 15cm.

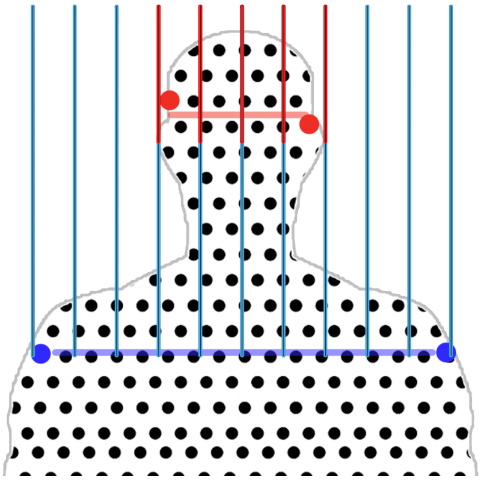


Figure 3.9: Figure illustrating the head width slices and chosen points in red and the shoulder width slices and chosen points in blue

Region features The region features define certain regions based around the possible head points that where explained in section 3.5.3. The regions are visualised in image 3.11.

- *Head box* and *Torso box*: Head box starts at the possible head point, goes 16cm horizontally in each direction and 30cm down. Next is the Torso box that starts 30cm below the possible head point, goes 16cm horizontally in each direction and another 30cm down.

$$\text{HeadBox}(x, y, z) = x_{head} - 16 < x < x_{head} + 16, y_{head} - 30 < y < y_{head}, z \in \mathbb{R} \tag{3.15}$$
$$\text{TorsoBox}(x, y, z) = x_{head} - 16 < x < x_{head} + 1, y_{head} - 60 < y < y_{head} - 30, z \in \mathbb{R} \tag{3.16}$$

- *Height*: the vertical value of Possible head point minus the value of Min Y.

$$\text{Head height} = y_{head} - min(y) \tag{3.17}$$

- *Upper box* and *Lower box* A line is drawn through the middle of an objects height to create a box above the line and one below the Line. These boxes are used to see if the amount of points are distributed. A human body is quite equally distributed with the points in both boxes in almost all standing positions. While sitting it can be significantly more in the lower box.

- *Center Box* A box being 20cm wide and 20cm long with its center at the centre mass point is made to see if the cluster is hollow at its center mass. If the center is hollow it is not a human.

Check if object is touching ground or obstructed This test is applied on both the SVM and geometrical filter. Due to performance reasons it is only applied on clusters that are recognised as humans by previous classification.

First test is to see if the cluster is in contact with the ground. This is done by comparing the *Min Y* of the cluster with where the ground is which need to be input manually when setting up the camera. If min Y is within of a 100mm from the ground it passes. If does not pass this another check to see if the cluster is blocked by another cluster which makes it seem not to touch the ground. If the cluster is not blocked and is not touching the ground then it will be classified as an object and not a human.

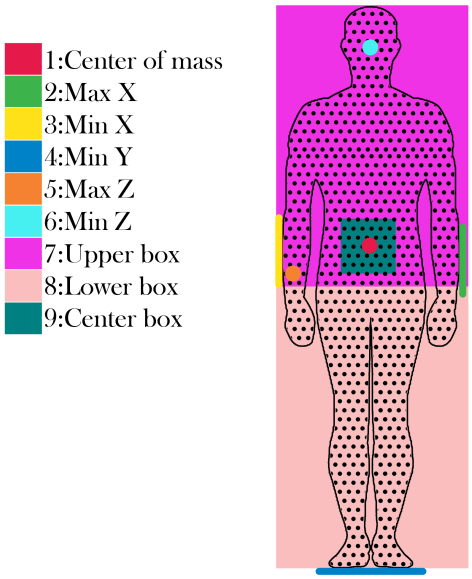


Figure 3.10: Geometrical features using the whole cluster

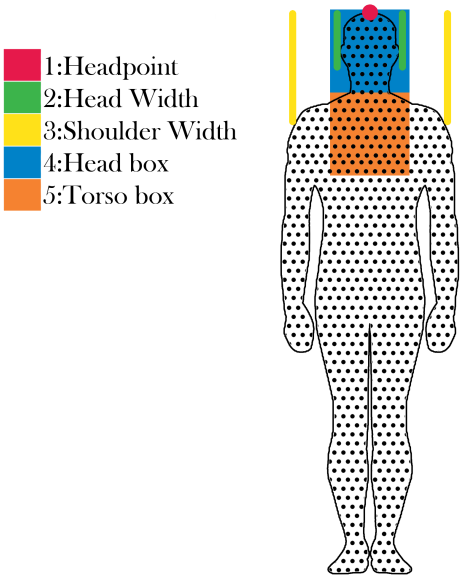


Figure 3.11: Geometrical features using the possible head point

Check object size This test is applied after both the SVM and the Heuristic-Geometrical classifiers. Due to performance reasons it is only applied on clusters that are recognised as humans by previous classification. The filter is heuristical in design to see if the human clusters have reasonable sizes.

First check is to see how much of the cluster width that is longer than 120cm and 90cm. The cluster is cut into slices of 20cm vertically and if five or more of them have a width of 120cm or wider it will be classified as non-human. It is also vertically cut into slices small enough to only contain one or two rows of data points using the principle from figure 3.4 and if 50% of these slices are 90cm or wider the cluster will also be classified as a non-human object.
It will also check how much of the cluster is touching the ground and taller than 180cm. As before it will cut into small horizontal slices and if the distance from the lowest data point to the highest data point in the slice is 180cm or more for 70% of the slices it will be a non-human. If all of the lowest data points, ± 2 to compensate noise, in the horizontal small slices is touching the ground the cluster will yet again be classified as non-human.

3.5.4 Heuristic-geometric filtering

In the first classifier named the 'Heuristic-geometric filter' the variables and region features presented in the previous section 3.5.3 and visualised in figure 3.10 and 3.11 are applied onto the clusters made by the clustering algorithm described in section 3.5.2. The thresholds in the following list are applied on the features, any object that pass all the thresholds is classified as human.

- The number of datapoints in *Torso box* must be more than the amount in Head box multiplied with 0.7
- The number of datapoints in *Head box* > 10
- The horizontal value of the possible head point minus *centermass* must be less then 800.
- Difference between highest and lowest horizontal value in *Head box* must be more than 150mm.
- The amount of datapoints in *Upper box* must be less than the amount in *Lower box* multiplied with 1.9.
- There must be more than 4 datapoints in *Center box*.
- *Max Z - Min Z* $< 1300mm$

- $10mm < Head\ width < 380mm$
- The difference between Possible head point in vertical perspective and *Min Y* cannot be less than 690mm
- Possible Head point is not allowed to have the same horizontal value as *Max X* or *Min X* plus minus the the minimum distance between two data points at that depth.
- *Torso width* > *Head width* OR *Torso depth* < *Head depth*.
- The amount of datapoints in the whole cluster must be more than 85.

Since this filter is heuristic in design and has been trained by iterative ad-hoc training it cannot be re-trained in the same way as a machine-learning based classifier. This makes it a more rigid filter with defined limitations as is discussed in section 5.2.

3.5.5 Feature extraction and feature vector design for SVM

In the second classifier the variables and region features presented earlier in section 3.5.3 and visualised in figure 3.10 and 3.11 are combined and used as features for a support vector machine (SVM). The theory of SVM is explained in section 2.4.

In total 15 features where used for the SVM. Twelve were combined out of previously presented region and variable features and three more were takes from paper by Luis E. Navarro-Serment et al[8]. A summary of all the features and their dimensions can be found in table 3.2.

The SVM uses a linear kernel and is trained on the ground truth data set presented in section 3.3. The data set was divided into training and testing data sets, with 1650 frames in the training set and 415 frames in the testing set.

SVM feature vector design The feature vector used by the support vector machine is presented in 3.2. It contain in total 34 entries making up 15 separate features.

Instead of using a threshold when comparing human geometric regions as in the Heuristic-geometric filter solution presented in the earlier section 3.5.4, some of the region and variable features presented are divided to give a relative relationship between the two regions that can serve as a feature.

The 'flatness' SVM features in table 3.2 are meant as a measurement of how flat one side of an object is. E.g all sides of a box appear very flat, or any large object placed directly on the floor would appear to have a very flat bottom viwed from the front. This flatness measurement is retrieved by slicing the object vertically. The height of the lowest point in the cluster is subtracted from the lowest point in each slice. Each of these height differences are summed and divided by the total number of slices. If the object in question is e.g a box it will have a completely flat bottom part and this number will be zero (or close to zero attributed to some noise). The exact same procedure can be modified to look at right-most, left-most or top-most points yielding flatness in other directions.

The moment of inertia and 3D-covariance matrixes are both proposed as features by Luis E. Navarro-Serment et al. [8] Calculating covariance and moment of inertia for 3D data yields 3x3 matrixes but since they are both mirrored, only the six unique values from each is used for the SVM feature vector. Slice features are proposed by K.Kidono et al[19].

3.5.6 Tracking

Noise can cause flickering for the detection, where a human that where discovered in the previous frame can not be in the next because of a 'spike' in noise. To counter this a tracker was constructed, remembering a humans last position. The main function being that any human has to disappear for two frames before the system acknowledge them as gone. In figure 3.12 an overview can be seen of how the tracker function works.

Human positions in the current and previous frames are compared using their centre of mass(x,y,z). Each

Support Vector Machine feature vector		
Feature number	Feature description	Dimensions
f1	Dynamic cluster size	1
f2	Thickness in depth axis	1
f3	Head width	1
f4	Torso width / Head width	1
f5	Head depth / Torso Depth	1
f6	Object absolute height	1
f7	Vertical distance, headpoint to center of mass	1
f8	Points upper box / points lower box	1
f9	Flatness top side	1
f10	Flatness bottom side	1
f11	Flatness right side	1
f12	Flatness left side	1
f13	Slice features	10
f14	3D-covariance	6
f15	Moment of inertia	6

Table 3.2: The different features included in the SVM and the number of variables in them (the dimension)

human position is also assigned a counter that increases with each frame, keeping track of how many frames ago the human was observed at that position.

When humans have been detected in a frame their centre of mass are calculated, we call these the human 'positions' (marked black in the figure). These positions are then compared to human positions from the previous frames (marked green and blue in the figure). The aim is to remove duplicates - i.e positions describing the same human in current and previous frames, while keeping old human positions that for some reason are not discovered in the current frame.

The old positions and new positions are compared and if they are within 150cm in real world distance, they are deemed to be the same person. If there are several within range, the closest one is deemed to be the same person. The old position is then removed in favour for the new one. The 150cm was chosen through ad-hoc iterative changes as the distance that gave the largest increase in true positives, while in reasonable distance for how far the centre of mass could reasonably move in the 500ms since last frame.

The usefulness of the function stems from when an old position can not be removed in favour for a new position. This old position likely then represents a human that for some reason (likely noise) was discovered in the last frame but not in the current one. The AGV should then in a real world scenario still act as there is a human at that position.

Lastly when moving on to the next frame, all positions get their previously mentioned counter increased by one, indicating how many frames ago they where discovered. For this implementation positions are only kept for two frames, if a new position is not found then, the position is removed.

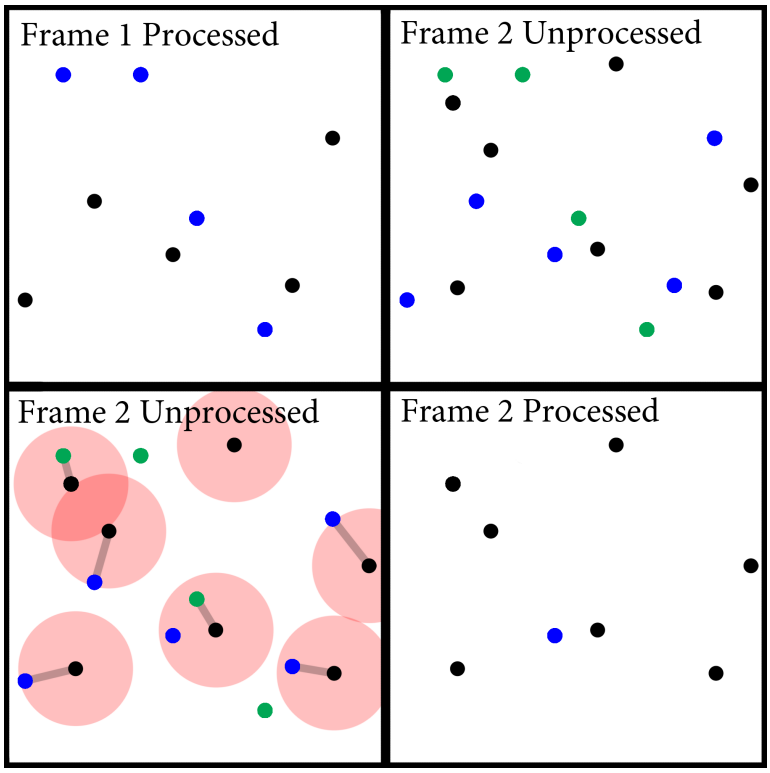


Figure 3.12: Illustration example of tracking algorithm. The colour of the location points indicate age in frames, black(new) = 0, blue = 1, green = 2. New and old points merges to their closest neighbour inside the search radius. Points that cant be merged and has counter > 1 are then removed.

4 Results

This first sections of this chapter provides the performance result of the human classification systems that are detailed in the methods section. Section 2.5 contains a literature study that densely summarises the findings of several studies of human interactions with autonomous vehicles.

4.1 Clustering algorithm performance evaluation

4.1.1 Time needed to run cluster algorithm

Regardless of the method used for classification (i.e. heuristic-geometric filters or SVM), the point cloud data first need to be clustered. In the first step of the clustering algorithm, all data points are compared to all other data points. This takes $O(n^2)$ time where n is the number of data points as n amount of points need to be compared n points where each comparison takes $O(1)$ time. After this all unassigned points will be compared to all data points which takes $O(n*u)$ where u is the unassigned points and u will always be less than n. While max u theoretically can be $u = n - 1$ it will realistically be significantly less than n. Last part of the algorithm will compare all master points to each other which takes $O(m^2)$ time where m is the amount of master points. The amount of master points can reach $m = n$ but it is unlikely and m will in most cases be lower than n.

This makes the time complexity for the the cluster algorithm:

$$O(n^2 + n * u + m^2) \tag{4.1}$$

and if it is assumed that $u = n$ and $m = n$ then it will instead be

$$O(3 * n^2) \tag{4.2}$$

This is the theoretical longest amount of time it can take to run through the algorithm. The number of data points going through the algorithm with the current camera will be between 0 to 25344 points. As seen in figure 4.1 the time increases exponentially.

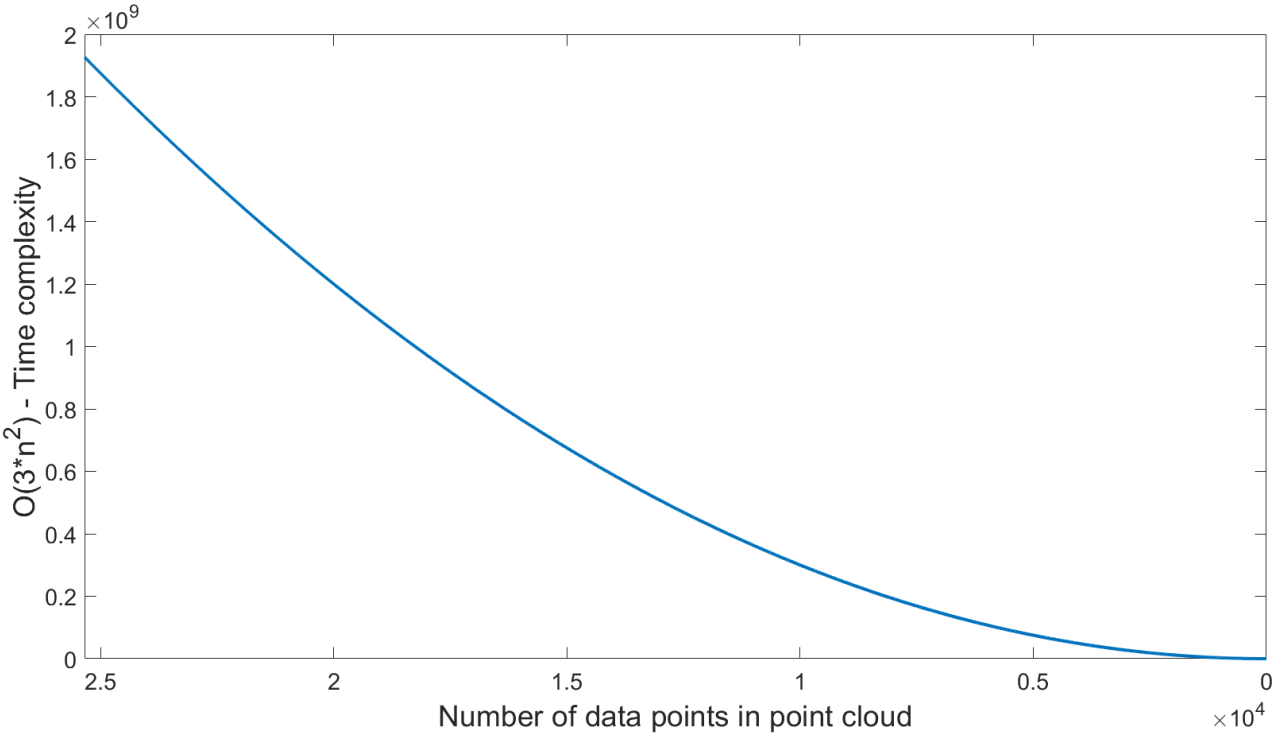


Figure 4.1: Graph illustrates how number of computations in clustering algorithm increase exponentially with increasing number of data points

Classifier performance				
Algorithm	Precision	Balanced Accuracy	Recall	SVM features included
Heuristic-Geometric Filter	0.6310	0.8990	0.8851	NaN
SVM	0.5689	0.7436	0.5218	f1-f12
SVM	No solution	No solution	No solution	f13-f15
SVM	0.2489	0.3339	0.9317	f1-f15

Table 4.1: Performance for the different classifiers, for SVM several different combinations of features are presented

4.2 Human localisation algorithm performance evaluation

This section presents the performance result of the two different classifiers, Heuristic-Geometric filters and SVM. The tests are performed on the data set presented in section 3.3. If a point cloud have more than 8000 data points then data points will be randomly removed until only 8000 remains, this is explained in section 3.5.1.

4.2.1 Effect on performance from loss of data

As presented in section 3.5.1 earlier in the report, if there are more than 8000 data points the full classification will need more then 500ms to run on the hardware. Tests were performed on the training data to see how removal of data points affect the results. The tests were performed by removing a certain percent of all the data at random in the point cloud and then running them through the whole algorithm. Since the removal is randomly distributed the test was repeated five times for each percentage and the mean is presented in table 4.3.

As seen in table 4.3 the algorithm yield a rise in FP after the first percent of removed data but after this performance is stable until between 20-30% removal of points where an increased decline of recall can be

Computation Time	
Algorithm	Seconds per frame
Geometry Filter Average	0.4636
Geometry Filter Fastest	0.4450
Geometry Filter Slowest	0.4992
SVM Average	0.4071
SVM Fastetst	0.3828
SVM Slowest	0.4610

Table 4.2: Computation time for the full image classification process, using the different classifiers. Averaging over 415 frames and five runs

observed, shown as a graph in figure 4.2. After this performance gets gradually worse.

It can also be observed how the computation time changes with the amount of data points decreasing. It can be seen when comparing the theoretical model of figure 4.1 and the results in figure 4.3 that the time required increases exponentially.

Remaining	True Pos	False Pos	True Neg	False Neg
100%	2362	466	12667	383
99%	2450	1162	15925	295
75%	2413	1302	15643	332
50%	1990	1186	14954	755
25%	834	792	14171	1900
10%	24	101	9045	2721
0%	0	0	0	2745

Table 4.3: The mean classification results for the training data with the geometry filter when randomly removing a certain percentage of the original data points.

4.3 Evaluation and tests of camera

The training data used for developing the system was all recorded with the Visionary-T V3S100-2AABAAB. Camera parameters are given in section 3.1. Before constructing the test videos, some experiments for the camera was set up to evaluate the camera and its limitations. Because of the challenge of measuring detrimental effects such as noise, the experiments are performed such that they can be used in a more general decision making on the technical limitations of the camera.

4.3.1 Distance and visibility

In the camera specifications in section 3.1, the camera is noted as having a working distance of 60 meters. A quick look at the camera output concludes that the camera has a high range but there is considerable noise on far distances.

The increasing noise with distance combined with the decreasing object resolution makes objects in the distance hard to distinguish even for the human eye. A live stream was set up to test at what distance this limit lies. In the test a human moved slowly away from the camera. This quick test resulted in an estimation that a human is no longer able to recognise a human somewhere between 10-15 meters or further depending on human pose in the picture. Here the cameras resolution is too poor for the human eye to accurately classify the object in the image. This result was used when setting the demarcation that an human object must be taller than 20px in the picture, for this camera 8,5m away. Limitations are discussed more in section 5.2

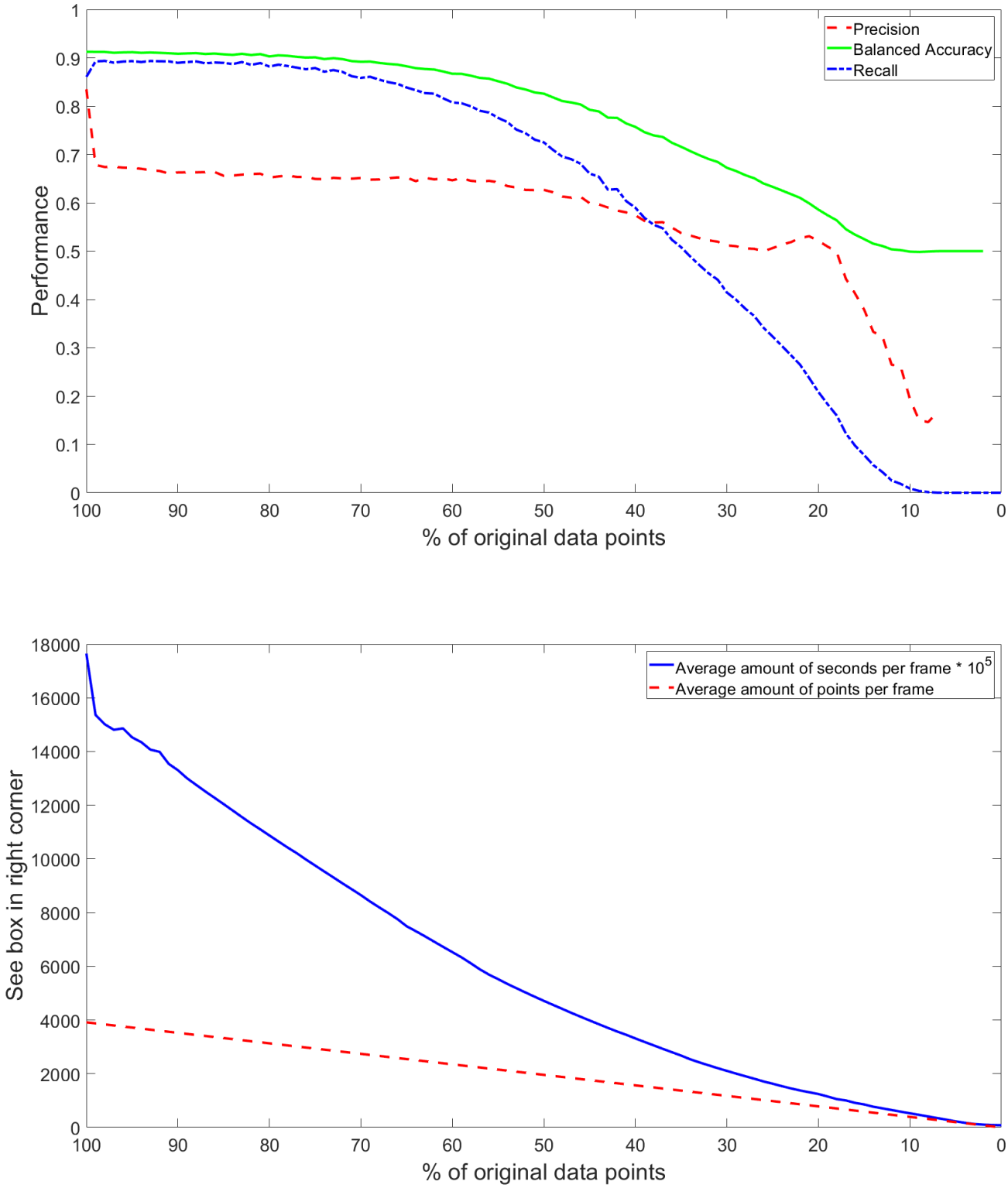


Figure 4.3: Plot showing average amount of seconds it takes to process a frame and average amount of data points in each frame for all percentages. Note that the seconds is multiplied with 10^5 to make the plot easier to read

4.3.2 Frame rate

In the camera technical specification document, it is said that the camera can capture 50 depth pictures per second at maximum. This is only possible when using specific settings in the camera. For the project other settings are used and to determine the feasible fps (frame per second) it can capture with the specific settings, a simple experiment was performed. The test was made with the camera directed at a big object, so all of the pixels had a value in them. The test were run in Matlab where in a loop run for 60 seconds where the cameras current frame was extracted and saved to an array with a pre-defined size to save time. from the average of 6 runs, it was able to go through this loop 8793 times in 60 seconds. Therefore, the camera was able to extract a picture and save it 146 times per second, almost three times as fast as the optimal capture speed of the camera. The amount of times it is able to do this is dependent on the hardware and its current state(i.e temperature and what tasks it is doing)
Next part of the test was concerning the number of unique frames among the saved frames. of the saved pictures that was unique. This was done by going through every picture, except the last, and comparing it with the one in front of it. For all 6 tests it found 750 unique pictures out of on average 8793 saved pictures. Therefore, in 60 sec, the camera was able to capture 750 pictures. Considering this, with these settings, the camera can capture 12.5 pictures per second. The requirement to be able to process a frame in less than 500ms from section 3.4.1 is not compromised by the frame rate as it can capture a frame in 80ms, which is lower than 500ms.

Time	Saved pictures	Unique pictures	Frame rate
60	9089	750	12.5
60	8788	750	12.5
60	8728	750	12.5
60	8776	750	12.5
60	8585	750	12.5
60	8793	750	12.5

Table 4.4: Results from camera frame rate tests

4.3.3 Interference from daylight

Measuring detrimental effects such as noise is a difficult task. But after running the camera in the same area on cloudy and sunny days, it can at least be confirmed because of increased IR (infrared) content on sunny days. In the technical details for the camera, it is said that its light sensitivity is below 50000 lux which is in the range of direct sunlight[28].

4.3.4 Effect of reflective surfaces

A test was set up where a human wearing a reflective vest walks through a corridor and in an open space. The result from both showed that the Visionary-T V3S100-2AABAAB has serious problems with correctly capturing the scene regardless of the internal filters turned on. Multiple artefacts form around the human showing in the point cloud as swirling streaks around the upper body impacting the shape of the data cluster that was supposed to represent the human. Figure 4.5 and 4.4 shows the difference between a human wearing a reflective vest and without. Less serious but similar noise occurred from very shiny surfaces that reflected natural light from the windows.

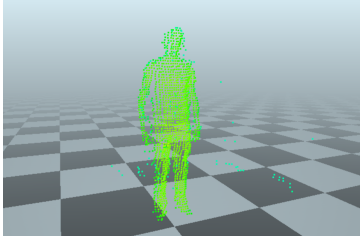


Figure 4.4: Human.

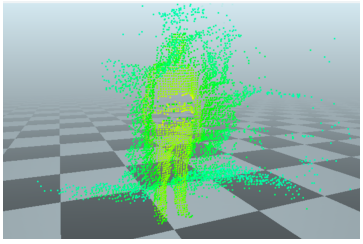


Figure 4.5: human with reflective west.

4.4 Interview about AGV behaviour

A deep interview was performed at Nolato utilising AGV solutions in Göteborg with a supervisor about their and the staff's experiences using AGVs in a warehouse. The conclusion from the interview was is summarized as following: :

- The most major concern was collisions between AGVs and manually operated forklifts. The AGVs were unable to detect the high forks of the other forklift blocking the path causing a collision because the AGVs detected only the floor space ahead as free.
- Noise levels from the AGVs were something the employees found annoying and more or louder sounds were something they wanted to avoid. Beeping sounds were expressed to be especially unwanted.
- Unfamiliarity with the AGVs was a concern for new employees as they did not know how they worked or reacted to different scenarios, but they learned with experience.
- When problems arise, for example when the AGV stops moving, it is hard for the nearby staff to understand why and what can be done about it.

5 Discussion and Conclusion

The discussion section will start covering the performance of the classification system for recognition of the human, based on 2D-range ToF data. Next, a proposed Human-Machine-Interaction(HMI) design involving signals and warning for human-AGV interaction is detailed. Lastly, a conclusion summarises the major contributions of this thesis project together with proposing future research on the subject.

5.1 Evaluation of camera

After several tests with camera, certain characteristics of the camera were discovered that should be taken into consideration when designing a safety system. Reflective areas had major disruptive effect on the camera output as data points smeared the original scene and introduced many artifacts. If the hardware used to capture three dimensional data cannot deal with this properly, it will cause major problems if employees are wearing reflective work vests. The errors caused by this can clearly be seen in figure **??** and **??**.

This specific issue with reflective surfaces could be solved with a 3D camera not using Time of Flight as light hitting the reflective surfaces gives unreliable results with the ToF camera. A stereoscopic camera would likely not have this issue, however there are a large number of other factors that need to be taken into account.

The technical specification document of the camera states that it would function up to 60m radial distance. While it is able to measure a single point within that distance, the presence of high noise becomes very noticeable for many objects in the distance as their pixel density decreases. It just took around 8 meters before even the humans started to have difficulty in identifying the human-objects in 3D point-cloud. The low resolution of the camera combined with the detection angle, seen in section 3.1, made objects at a distance consist of very few data points compared to objects close up. If point clouds are to be used effectively at longer distances for recognising objects either the resolution needs to be increased, simultaneously increasing computational needs, or the angle needs to be decreased. However, for a safety implementation, the angle cannot be too small as this would not give the AGV a wide enough field of view to detect humans approaching from the side.

In the point clouds recorded by the camera, objects cast a shadow behind them that could cause false positives on walls. The prevalence of this problem could increase with a larger number of pedestrians, it would have to be tested.

5.2 Project limitations and their effects on the final result

For creating a heuristic-based geometrical filtering algorithm, some generalisations had to be made on the human form. The goal for a human identifier is of course to be able to recognise humans in all possible situations but the techniques used for the proposed classification algorithms come with a few limitations on what possible scenarios a human can be discovered in. Some of these were set early in the project, some were removed or set less strictly as the system performed better. For the final system if the following are fulfilled, the aim is for the system to able to classify the human.

- From human head point and at least 700mm downwards of the human must be visible
- Human head must be reasonably above the torso
- Human head reasonably head-shaped e.g wearing a sombrero would not work.
- Humans carrying objects are assumed to be carrying them reasonably balanced
- Items cannot be carried directly above the head otherwise the head cannot be located
- A human can not be too much in contact with a large object, e.g full body leaning against a wall or it causes problems for clustering
- Human outline can not be too undefined, in practice this means human height $>$ 20 pixels/points

From these demarcations, one can quickly see that for example a human lying horizontally on the floor would not pass all these criteria. It is however noteworthy that this simply means that the human object will not be classified as a human. The clustering algorithm will still recognise the human as an object so collision can be avoided. These only represents the current limitations of the system, with further work and extra features some of these limits as head above torso could possibly be relaxed.

The limits were taken into account when recording the validation set of labelled data. All videos were recorded so that poses or distances involved in scenarios would not be outside of the well-defined limitations of the system.

5.3 Evaluation of clustering

5.3.1 Cluster algorithm

Using clustering in 3D space is useful when identifying the objects and extracting features for the identification and recognition. The downside as it could be seen in figure 4.1 in the result chapter is that it becomes less useful for time critical objectives as more data points become available. To combat this, either more powerful hardware would need to be used or stricter pre-processing which can remove more data points. An example of stricter pre-processing could for this particular camera be adjusting the region of interest even stricter by setting limits where humans are not needed to be detected. At the moment only points at the roof and floor are removed. A hard limit for how distant points can be in the horizontal and depth perspective could be implemented.

The future will likely bring new 3D cameras with higher resolution than the ones currently used today. It is not a stretch to say it is inevitable that with the higher resolution a better classification algorithm to identify humans could be designed. But with higher resolution a more effective clustering algorithm would be needed in tasks that are time critical. This could also be solved by using better hardware for running the algorithm if the algorithm cannot become more effective.

This particular cluster algorithm needs to have the floor removed to properly cluster and objects in racks will not be classified correctly and they would simply be included in the clusters belonging to racks. For clustering objects standing on the same plane, this behaviour is not a problem but for alternative uses of the clustering algorithm this could pose challenges.

Shadows in the point cloud may become an obstacle in front of achieving good clustering performance. Because a single object can cast a shadow on itself and then cause itself to be divided into several clusters since these data points may not be close enough to each other to be considered as a coherent cluster. This observation can be seen in figure 5.1.
A shadow-casting object can cause objects behind it to lose their properties connected with identification and the over-shadowed objects may show up as broken patches of misidentified objects.

5.3.2 Cluster versus Object problem/dilemma

In this report, an 'object' is referred to both humans and inanimate objects such as boxes, racks or other forklifts. For a computer working with a point cloud there is currently no simple and none-heavy computational way to separate real world objects, instead clusters have been used. A cluster have been defined as data points in the point cloud being in the same 3D proximity which can be assumed to form a coherent 3D structure and can be expressed in a clear mathematical definition using 3D Cartesian coordinates and their neighbourhood connection matrix. This works well for most cases since all data points close to each other would belong to the same object but there are still cases where this assumption may not be true. Sometimes two distinctive objects can be clustered as one object since they may be too close to each other. The opposite problem exists as well, if the data points belonging to an object is too far apart for some reason, such as shadows from an object in front, they might get divided in several clusters. As seen in figure 5.1 where a human is holding a piece of paper that is casting a shadow. This might cause the clustering to split split the object corresponding to human into two or more clusters.

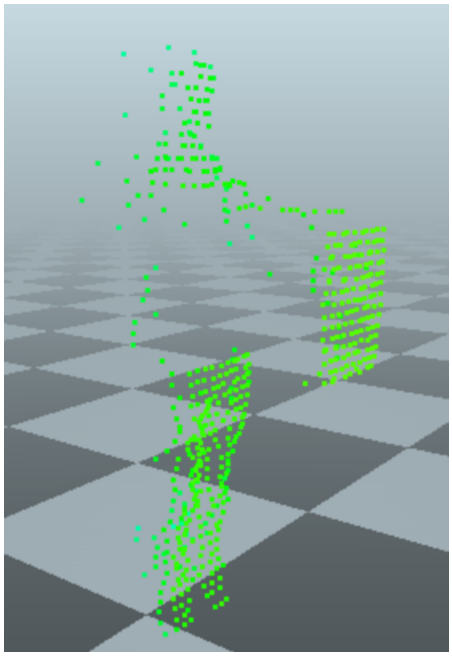


Figure 5.1: Human holding a piece of paper and casting a shadow on himself

5.4 Evaluation of algorithm

The gradual loss in resolution is tightly connected with decreasing the range where algorithm can perform acceptably. Since neither the algorithm nor a human eye proved accurate at classifying humans if they are less than 20pixels tall. Much higher resolution than that does not however seem important for the classifier, and less data points mean faster processing. In this paper, we used a random process for removal of excess data points but there is much room for improvement on data size reduction. Since clustering is the most processing-heavy part, an algorithm that could, in a quick and reliably way, strip down the size of objects close to the camera, while keeping those in the distance would give a major decrease in computation time. While there is some indication to have some effect on performance theoretically, in practice we have observed no significant effects.

The classifier using Heuristical-geometrical filters, which can be considered as a white-box (i.e. transparent) algorithm since all its functions can be understood, were tuned using ad-hoc approaches and heuristics on-the-fly based on the training data to explore the room for improvement. The initial experiments helped to identify two opportunities for improvement. First, is to have a larger set of data to avoid overfitting or missing out on situations that can prove hard for the algorithm to solve and gives inspiration for defining and extracting new features to make it better. The second is to move beyond the ad-hoc and heuristic approach towards a more systematical design strategy by enforcing it by testing all possible combinations, with restrictions to limit run-time, until an optimal result is found.

The proposed classifiers are only using the coordinates in the point cloud to detect humans but there is more information available that could increase the performance. With the ToF camera it is possible to save information such as intensity and confidence and there is a possibility that these could be used for features to increase the performance of the algorithm. RGB data could also be used together with the depth data for extracting colour based features. An RGB image must however be taken with a separate camera and aligned since ToF cameras do not take colour photos, RGB-D approach could therefore mean a noteworthy amount of extra processing.

5.4.1 Minimising false positives or false negatives

While designing the classifier a clear definition of solution approach to the problem need to be chosen. It must be decided whether all the objects that are found in the frame should be classified as 'human' until it is disapproved or not. This choice of assumption in the algorithm mainly affects the number of FP and FN in the results. It must be decided whether all the objects that are found in the frame should be classified as 'human' until it is disapproved or not. This choice of assumption in the algorithm mainly affects the number of FP and FN in the results.

The classifiers presented in this work were designed with the safety in focus, rather than improving the TP rate or decreasing the FP. The large number of FP do not make the system 'unsafe', however it may render the operation of the AGV cumbersome since it will give a large number of false warnings. These warnings may become 'annoying' for the warehouse workers as well as slowing down the work-flow.

5.5 Evaluation of the data set

Since all of the training data were collected on two persons, it is a risk that the algorithm is over-trained and would give less successful results on people with different body shapes. The same could be said with the lightning conditions as all the tests took place in two different premises during varying times of the day.

Background objects could not be properly tested as there was a limited amount of objects to use in the background during tests. In an attempt to prepare an adversarial attack on the algorithm, objects are stacked in human-like shapes. It was observed that it was possible to trick the system into doing false positives. If there would be objects with human like qualities like heads, shoulders, size and proportions it is a big risk of a false positive.

In this project it was a goal for the system to be able to detect humans in both standing and sitting positions. To achieve this, many different human shape templates were needed to be considered. If the system would no longer abide by this requirement, it could be more optimised for standing humans and reduce false positives and false negatives, at the cost of no longer being able to detect people sitting down.

5.6 Human AGV interaction

5.6.1 When to seek attention

The AGV should not demand the attention from humans unless needed to avoid being distracting or disturbing. An AGV should seek attention when:

- It can increase safety, such as when a human is in its way.
- It can help a human make a decision by showing which direction the AGV will go or what kind of action it is planning to make.

5.6.2 Signalling attentions for safe passage

As a safety system, it is important for humans to quickly and clearly understand the meaning from the signal produced by the AGV. This has two purposes; the first is to dissuade the humans to take an action that has a high risk of putting the human in the AGVs path, for example jumping out behind a corner or reducing efficiency by standing and blocking the AGV's path. The second purpose is to make humans feel more comfortable working with the AGV, by avoiding any actions that may cause startling, annoyance or unpredictability.

From tests previously done with cars mentioned in section 2.5.2, it seems that people feel safe enough to cross the road when they have a clear sign that they recognise, such as traffic light showing who have right of way, and when they feel that they are noticed. This communication could be either by getting eye contact with the driver and a nod or the car slowing down. This can be used by an AGV to inform staff of its intention. Instead of focusing on a way to show it is safe to pass by, a system could tell the staff they are being seen by

the AGV. This could potentially be more effective as it would intuitively tell the staff it is safe if they know that the AGV is supposed to stop when they can see people and they are too close. One way this system could work would be by pointing a laser pointer or lamp at the person it is seeing. This would make it clear when the AGV sees someone and equally as important, when it does not see someone. With an indoor environment it would be no interference from sunlight that could make a laser or lamp difficult to see.

Sounds to mark presence

If the recommended standards are to be followed for warning sounds in electric cars as presented in 2.5.3, the AGV should make a constant sound that is easily identifiable and not annoying. It should have a distinct sound for constant speed, acceleration, deceleration and for the conditions of 'being blocked by an obstacle' so it can not move. The volume should change depending on circumstances and adjust to the environment. In a loud environment, the volume will increase so it still can be heard and when approaching a corner. The volume needs to be increased so to make sure a potential person round the corner will know of the AGV and not walk out in front of it.

For this, the sound of a combustion engine could be used. Studies concerning human behaviour when interacting with vehicles and mobile platforms (see section 2.5.4 and 2.5.2) have shown that the humans react well to shows that human react well to what they are already familiar with or have learned. This would be an intuitive way for humans to know if the AGV is accelerating, decelerating, having a constant speed or starting up. Engine sounds would not be as annoying as a beeping sound which could be reserved for emergency sounds.

5.6.3 AGV behaviour

Manoeuvres

If the AGV can only fulfil the basic safety standards the interaction behaviour would be rather rigid. When an AGV detects an obstacle, it will stop in front of it until the obstacle is removed. Moving around it would require to step off the planned path and 'improvise' with the inputs it has. As of today the minimum surrounding awareness needed for an AGV is a bumber or non-coontact sensor at floor height. Manoeuvring around the object would be hard if not impossible to do without more advanced sensors and motion planning algorithms than the minimum needed.

With additional inputs such as those from a 3D camera, it would be possible to know the immediate surroundings to evaluate if the maneuver would be possible and also initiate it before the AGV is in front of the obstacle so the manoeuvre is executed smoother. If the AGV knows what kind of object it is, it can choose appropriate action. A human is expected to move around and therefore it is risky to move around since the human trajectory of motion might be unpredictable. If the obstacle instead is a stack of boxes, there is no unpredictable movement and manoeuvring around it can be a possible course of action.

Blind spots

When using a device at ground level to detect obstacles, objects in the air can be missed. It was said in the interview presented in section 4.4 that there had been accident with AGV driving into the forks of another forklift. This problem was solved by giving the AGV right of way but similar situations can be avoided by making the AGV able to see the whole space in front of it.

Using a 3D camera or another 3D scanner, corners and blind spots could be known and the speed of the AGV could be adjusted to be able to handle a potential emergency brake. If no blind spots could be found, such as if it drove in a corridor with solid walls, then the AGV could increase the speed as it knows there is no risk of sudden obstacles.

5.7 Ethical concerns

While some argue that there are less ethical concerns when working with automation than other areas of engineering, a vital role of any engineer is being able to critically evaluate their own work. This should stretch

beyond simple safety analysis to address ethical consequences. Following two sections contain the authors reflection regarding the ethical concerns of automation.

5.7.1 Camera recordings and privacy

When equipping an AGV with cameras capable of recording, privacy concerns for the employees will arise since it makes it possible to record them in the workplace. This could be addressed by automatically or systematically deleting the recordings from the cameras, so that no footage of the employees is saved. However, if despite of safety procedures, an accident occur, footage can be reviewed to analyse what caused the accident and improve safety. Deleting the recordings immediately would then be detrimental for the accident reviewing process. Perhaps a similar system like the EDR(Event data recorder) used in many new cars could be a solution. In such a device, the recent video data can be overwritten and only archived in permanent memory after accidents. Another possibility is if data can be processed by the system and then delete the original recordings leaving only meta-data that can not be tied to a specific individual. As with other recordings, owners of the system would nonetheless need to sign consent forms to acknowledge on how these recordings may be used.

5.7.2 The downside of automation

Every change that could bring something negative, be it cost, environmental impact or human inconvenience should be carefully weighed against the positive effects.

AGVs reduce the manpower needed for certain tasks and will probably also decrease the need for employees in certain areas. This could lead to people being laid off. Unemployment is always a burden on the individual which is why most countries have laws regarding how staff can be laid off. However, if the automation process of a large workplace is very rapid many people in the local community might end up unemployed which could possibly have a detrimental effect on the whole local society.

Automation is generally cheaper in the long run than manpower but laying people off purely for company monetary gain is hard to motivate ethically. One common and accepted reason is that if there is a significant risk of even more people losing their job if not company expenditure is reduced. However with the steep initial investment cost needed for most automation this is hardly applicable. A different reason specifically for automation of fork lift work and similar tasks would be that there is a huge number of accidents worldwide associated with fork lifts. According to both the German and American accident databases, in almost half of the accidents that resulted in a death the driver was killed. If automation could drastically reduce injuries and deaths, then perhaps the loss of jobs in the area of fork lifts could be motivated.

The responsibility of ethical introduction of automation should ideally not be left on one part. Both the company supplying the automation as well as the buyer/employer would do well in taking responsibility and making informed decisions. On the sellers part, they need to acknowledge the possible lay offs their product might cause, and therefore as previously mentioned make the safety of their product a top priority. Perhaps even propose gradual automation to avoid massive lay offs. On the buyer/employers part lies the previously mentioned responsibility of not performing ethically questionable lay offs. Possibly performing in-house education to move the employees to other parts of the company. Or step-wise introduction of automation giving laid off staff compensation and time to find new work. As previously mentioned the strongest reason for automation being ethically defensible is the increase in safety, so the employer should always install automation with safety being the first criteria, monetary/efficiency gain coming second as long as keeping in budget.

Automated machines and people working in the same space of course also pose a risks for the employees. Accidents will happen if the systems are not good enough at identifying risks or employees are not educated in how to act around the automated machines. One current solution in several factories for decreasing the risk of serious and fatal injuries is to establish fully automated or restricted zones where people are not allowed to roam and automated machines can function without the risk of people being in the way. Despite this precaution the responsibility of a serious accident cannot be completely unloaded from the developers. The safety goal should always be making machines more aware of their surroundings.

5.8 Conclusion and future research

The results show that 3D cameras have potential as tools for human classification even at low resolution. It also shows great promise in finding obstructions of all kinds. An AGV should be able to perform the calculations even under time critical conditions, given smart pre-processing. Furthermore this should be possible without high-end hardware.

5.8.1 Algorithm conclusions

The Heuristical-geometrical classifier did not give a result that would be useful for real world application in the current state on the validation data but it seems possible to come up with new features that could enhance it. For fine tuning the Heuristical-geometrical approach will need some sort of gradient decent or similar mathematical minimising algorithm to tune the thresholds. This will however likely not bring down the false positive rate drastically, so more features would have to be added. The authors believe that more features and also an analysis on which one of these features carry minimum redundancy and maximum relevance could make a difference based on the fact that the false positives comes from the same objects being miss-classified in each frame and not at random, an example being a truck in the background of the validation data that gets classified as a human if seen from behind. These miss-classified structures could be analysed and appropriate features addressing them could be added.

From the Heuristical-geometrical approach, it can still be seen that a remarkable amount of filtering can be achieved with only ad-hoc anthropomorphic filtering when using Cartesian data. Leading to the conclusion that there is a strong possibility that a high performing classifier could be constructed.

The SVM application of the geometrical features seem to be just moderately successful, with a relatively high rate of false positives. SVM has a tendency to perform worse if they contain irrelevant features, so the reason might be that some features need to be removed. It could also be a problem regarding that human 3D geometry appears very different when standing, crouching, stretching and carrying objects. There might be an issue currently where the variance of some features become to large for the human category which makes the SVM not capable of finding a proper hyperplane. A very strong contender for future research would be trying out a non-binary classifier. Separately labelling and training on standing, sitting, crouching humans to involve multiple poses. Nonetheless the result point towards that geometrical features might be applicable as SVM features.

Finally, using only a point cloud with Cartesian coordinates as data to classify humans will always make it impossible to differentiate between an object with the shape of a human and a 'real' human, the extreme scenario would be e.g a mannequin. However adding extra data as intensity or colour to each data point so materials can be distinguished could prove valuable to raise performance. K.Kidono[19] that proposed the slice features also proposes using "Distribution of the reflection intensity, which is composed of the mean, the standard deviation and the normalised 1D histogram". This could perhaps prove a worthwhile addition to the current features.

5.8.2 AGV behaviour conclusion

Today AGV behaviour is quite simple and their input from the surrounding is limited. The safety standard only requires short range scanners along the floor that detects obstacles.

An important step in designing more advanced safety behaviour is more input data than the current standard. This enables design that allows the AGV to independently make safety decisions as adjusting speed when approaching a risk zone, such as a corner, or navigating around stationary blocking obstacles instead of standing still until they are removed. An optimal AGV-human interaction system is not only is safe but creates a feeling of safety. This is exemplified with an AGV that can clearly show it sees each individual, in opposition to a system that e.g only shows when it is safe to pass.

5.8.3 Future research

In this report, only a ToF 3D camera have been used to collect the point clouds needed. The algorithm should potentially work on other devices capable of creating point clouds and this should be explored. The code algorithm should fit devices that collect data from all around them and not only in front of them as it currently does.

Future work for the Heuristic-geometrical solution is listed here:

- Try splitting the test into several cases involving different human poses. Now there are only human or non-human classification options. Extending the options to e.g standing human, siting human, crouching human, human carrying something and non-human would make it possible to have stricter criteria. Most features could be reused for the different cases but with different more scenario specific thresholds.
- Making use of the intensity and/or confidence data that the ToF camera could provide was considered but never implemented. Intensity vary with different materials so it could potentially make an important difference between clothing and other materials.
- A system to auto adjust variables in the algorithm like gradient decent would be guaranteed more or less beneficial but the SVM approach was choosen. The thresholds where now just updated by an ad-hoc procedure after running on the training data.
- Using a weighted system to decide if a cluster is human would have an impact on false negatives. For the geometrical filter it now needs to pass all the tests to be classified as a human. With a weight system more tests could be added and be weighted differently depending on how reasonable it is to fail the test and still be a human.
- Label data in 3D, for this project all data were labels were made in 2D data using built in functions in Matlab. It gives the correct number of TP and FP but in some cases e.g when clustering algorithm fails and splits an object it can be hard to know which part of the split it thought was the human.

Future work for the SVM:

- Try out a non-binary SVM classifier where crouching and standing humans etc are two different categories, decreasing the in variance of certain features between the objects in the same group.
- Try utilising object intensity as a feature, all ToF cameras can output intensity as well as depth for each data pixel.
- Apply some SVM feature selection algorithms to further sort out what geometrical features are relevant for the SVM.

References

[1] G. Anderson and D. Plecas. The science of warning lights. *The Journal of Criminal Justice Research* **1** (Jan. 2010), 10.

[2] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh, Pennsylvania, USA, 1992, pp. 144–152. DOI: 10.1145/130385.130401.

[3] R. Bostelman, T. Hong, and R. Eastman. "Safety and performance standard developments for automated guided vehicles". Sept. 2014, pp. 487–494. DOI: 10.1142/9789814623353_0057.

[4] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**.6 (Nov. 1986), 679–698. DOI: 10.1109/TPAMI.1986.4767851.

[5] M. Clamann, M. Aubert, and M. Cummings. Jan. 2017.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning* **20**.3 (Sept. 1995), 273–297. DOI: 10.1007/BF00994018.

[7] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. June 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[8] L. E. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian Detection and Tracking Using Three-dimensional LADAR Data. *I. J. Robotic Res.* **29** (Oct. 2010), 1516–1528. DOI: 10.1007/978-3-642-13408-1_10.

[9] A. J. Elliot. Color and psychological functioning: a review of theoretical and empirical work. *frontiers in Psychology* (Apr. 2015). DOI: 10.3389/fpsyg.2015.00368.

[10] M. Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". *KDD*. 1996.

[11] U. N. E. C. for Europe. *Proposal for guidelines on measures ensuring the audibility of hybrid and electric vehicles*. Mar. 2011.

[12] S. Fiore et al. Toward understanding social cues and signals in human-robot interaction: Effects of robot gaze and proxemic behavior. *Frontiers in psychology* **4** (Nov. 2013), 859. DOI: 10.3389/fpsyg.2013.00859.

[13] I. T. S. D. Foundation. *Safety Standard for Guided Industrial Vehicles and Automated Functions Of Manned Industrial Vehicles, ANSI/ITSDF B56.5-2005*. 2005.

[14] S. Francis et al. A ToF-Camera as a 3D Vision Sensor for Autonomous Mobile Robotics. *International Journal of Advanced Robotic Systems* **12** (Nov. 2015), 1. DOI: 10.5772/61348.

[18] M. H. Khan et al. "Multiple human detection in depth images". *18th International Workshop on Multimedia Signal Processing(MMSP)*. Montreal, QC, Canada, 2016. DOI: 10.1109/MMSP.2016.7813385.

[19] K. Kidono et al. "Pedestrian recognition using high-definition LIDAR". *2011 IEEE Intelligent Vehicles Symposium (IV)*. June 2011, pp. 405–410. DOI: 10.1109/IVS.2011.5940433.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* **25** (Jan. 2012). DOI: 10.1145/3065386.

[21] K. Mahesh Kumar and A. Rama Mohan Reddy. A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method. *Science Direct Pattern Recognition* **58** (Oct. 2016), 39–48. DOI: 10.1016/j.patcog.2016.03.008.

[23] J.-F. Petiot, B. Kristensen, and A. Maier. "How should an electric vehicle sound? User and expert perception". *IDETC/CIE*. 2013.

[25] A. Rasouli and J. K. Tsotsos. Autonomous Vehicles That Interact With Pedestrians: A Survey of Theory and Practice. *IEEE Transactions on Intelligent Transportation Systems* (2019), 1–19. DOI: 10.1109/TITS.2019.2901817.

[26] *Safety of industrial trucks – Driver- less trucks and their systems, SS-EN 1525*. Nov. 1997.

[27] F. Schneemann and I. Gohl. "Analyzing driver-pedestrian interaction at crosswalks: A contribution to autonomous driving in urban environments". June 2016, pp. 38–43. DOI: 10.1109/IVS.2016.7535361.

[28] *Visionary-T*. Accessed on 2019-03-26. SICK AG. Waldkirch, 2018. URL: https://cdn.sick.com/media/docs/6/76/576/Product_information_Visionary_T_en_IM0077576.PDF.

[29] X. Wang et al. LBP-Based Edge Detection Method for Depth Images With Low Resolutions. *IEEE Photonics Journal* **11** (Feb. 2019). DOI: 10.1109/JPHOT.2018.2884772. URL: https://ieeexplore.ieee.org/Xplore/home.jsp.

[30] L. Xia, C.-C. Chen, and J. K. Aggarwal. "Human detection using depth information by Kinect". *CVPR 2011 WORKSHOPS*. Colorado Springs, CO, USA, 2011. DOI: 10.1109/CVPRW.2011.5981811.

[31] Z. Yan, T. Duckett, and N. Bellotto. "Online learning for human classification in 3D LiDAR-based tracking". *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2017, pp. 864–871. DOI: 10.1109/IROS.2017.8202247.

A Appendix

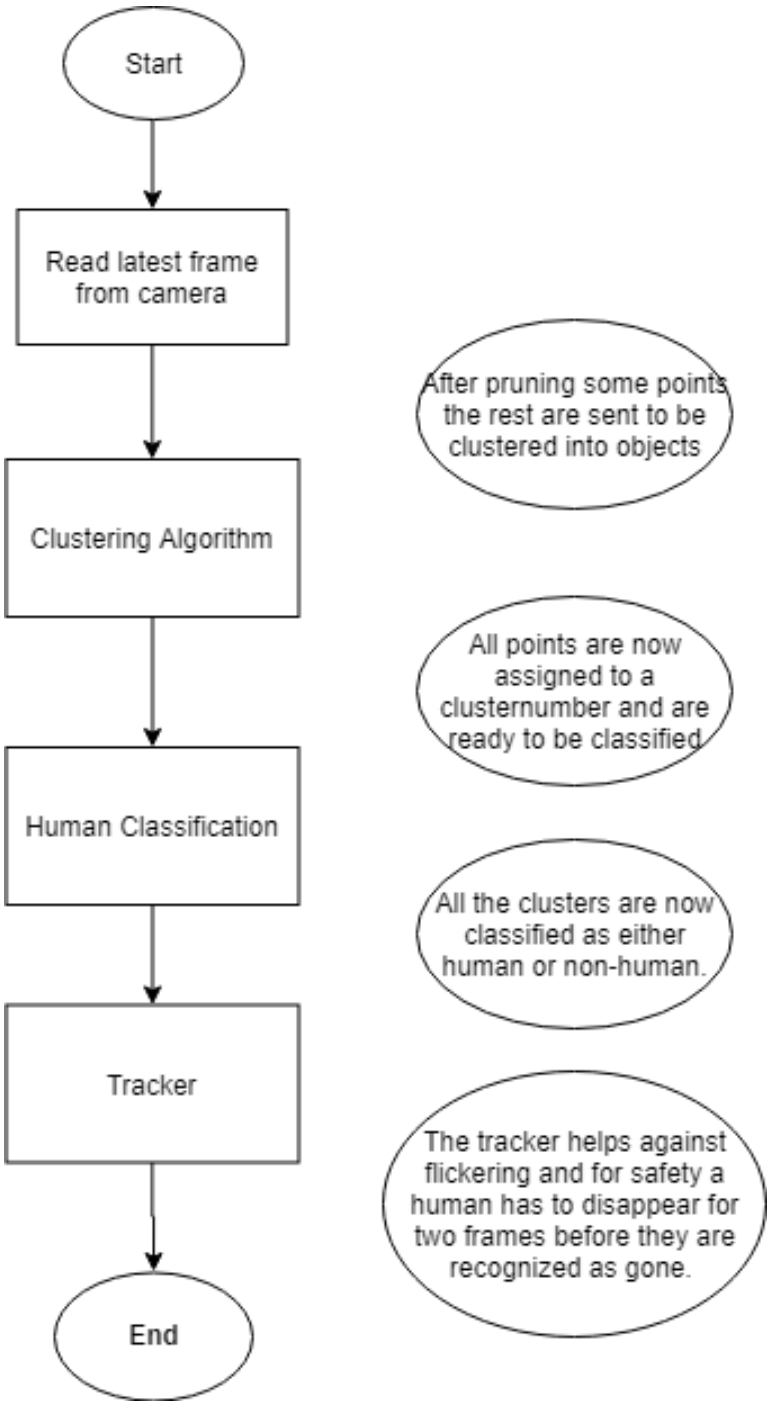
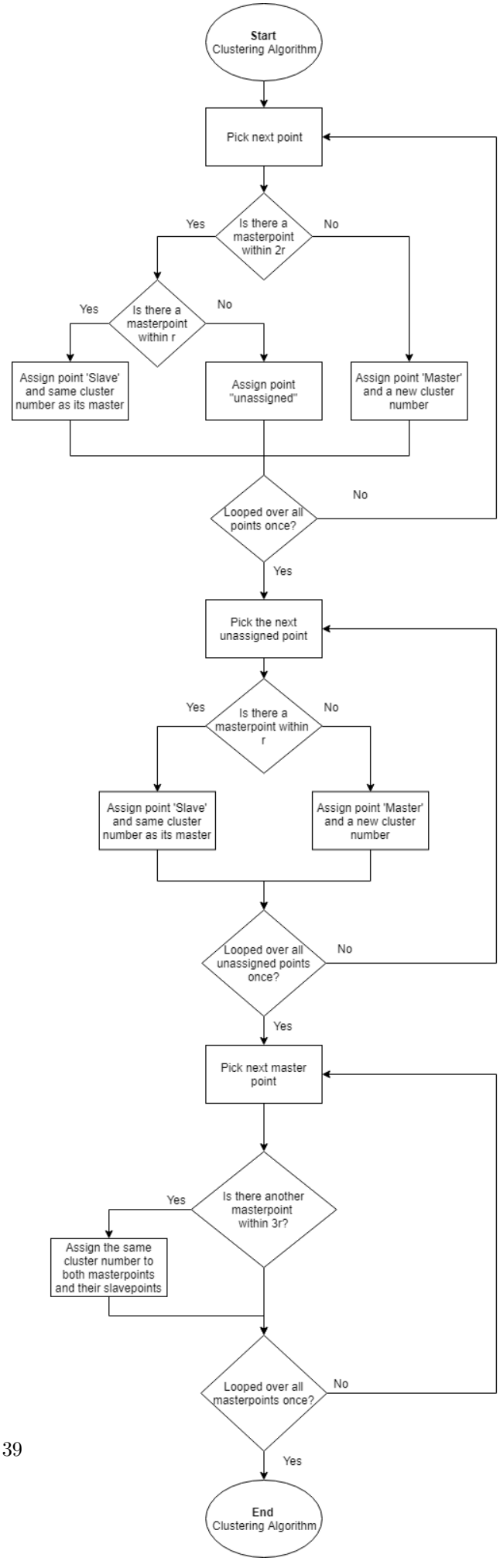


Figure A.1: Flowchart of the whole system. The clustering, classification and tracking are shown in depth in their respective sections


Start
Clustering Algorithm
Pick next point
Is there a masterpoint within 2r
Yes
No
Is there a masterpoint within r
Yes
No
Assign point 'Slave' and same cluster number as its master
Assign point "unassigned"
Assign point 'Master' and a new cluster number
Looped over all points once?
No
Yes
Pick the next unassigned point
Is there a masterpoint within r
Yes
No
Assign point 'Slave' and same cluster number as its master
Assign point 'Master' and a new cluster number
Looped over all unassigned points once?
No
Yes
Pick next master point
Is there another masterpoint within 3r?
Yes
Assign the same cluster number to both masterpoints and their slavepoints
Looped over all masterpoints once?
No
Yes
End
Clustering Algorithm