University of Hradec Králové

Faculty of Informatics and Management

Department of Informatics and Quantitative Methods

Computer Vision Applications in Advanced Driver Assistance Systems

Evaluation of stereo vision based object detection and ranging

Master's Thesis

Author: Bc. Jan Kučera

Study Programme: Applied Informatics

Supervisor: Ing. Karel Petránek

Hradec Králové 2016

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 2.5.2016 Jan Kučera

Acknowledgement:

I would like to thank Ing. Karel Petránek for his valuable comments and reviews regarding the thesis. I also would like thank my family for their patience during my writing sessions.

Annotation

This Master's Thesis deals with the evaluation of stereo vision based object detection and ranging in context to the Advanced Driver Assistance Systems. At first theoretical background on the various possible sensing techniques is presented with a special accent on the basics principles of stereo vision. Later an object detection and ranging algorithm is proposed that is immune to the noise that may be contained within a 3D point cloud. Further an automotive dataset that contains both stereo images and LIDAR data is introduced and measurements are done on top of it using LIDAR data as ground truth. About 3600 individual measurements are made to evaluate the performance of stereo vision and measurement errors are evaluated.

Anotace

Název: Aplikace počítačového vidění pro podporu řízení vozidel

Tato diplomová práce se zabývá vyhodnocením přesnosti detekce objektu a vzdálenosti s využitím 3D počítačového vidění v podobě stereo kamery. Na začátku práce je představen současný hardware používaný pro různé stupně určování vzdálenosti zejména v robotice. Dále je navržen detekční algoritmus pracující nad mračnem bodů, který je robustní vůči šumu pocházejícího z chyb v algoritmech pro stereo vidění. Je představen dataset zachycující dopravní situace obsahující stereo snímky i data z LIDARu, který je použit jako referenční měřidlo při určování vzdálenosti. Nakonec je vyhodnocena přesnost detekce objektu i vzdálenosti při použití stereo kamery.

Contents

Table of figures

Table of tables

1 Introduction

Technology is undertaking constant progress and the fields of robotics, autonomy and artificial intelligence are not an exception. Today's common consumer products like smartphones offer access to advanced solutions, many of which use some kind of artificial intelligence to increase their performance or user experience. With the increasing availability of embedded computing and precise perception sensors, robotics is undertaking fast development and is constantly settling new fields of human activity and new industries. With the increasing popularity of autonomous transporting solutions, mobile robots will pave the way to an increasingly autonomous future.

Mobile robotics is a very challenging field of research, because besides its intended task, a mobile robot is exposed to a dynamic environment which it has to understand and deal with. One of the elementary tasks of any mobile robot is to ensure the safety of itself and its environment. This task can be accomplished by the robot constantly monitoring its environment and making sure no harm or damage is made to anybody during its operation. An integral part of a safety system for mobile robotics is thus a collision avoidance system. To ensure collision avoidance while not relying on external information sources, the robot needs to be able to perceive its surrounding with highest possible resolution and in fastest possible manner to be able to take action in case it or its environment are in danger. Such perception is a technological challenge which has attracted a lot of research lately. A general collision avoidance system also finds its applications in safety systems outside the core robotics field with Advanced Driver Assistance Systems (ADAS) being an example here.

There have been numerous new automotive technologies introduced in the last couple of years that were focused on safety and that were based on the latest science and industry progress. The cars get smart and connected. Advanced Driver Assistance Systems have emerged as an extension to traditional driving safety technologies like airbags or ABS. The Advanced Driver Assistance Systems fall into the active safety group and employ advanced technologies to further enhance the overall driving comfort and especially driving safety. Precise environment sensing is needed for many of the ADAS features which is why there have been technologies brought from other fields and industries to be adapted for use in automotive applications, with radar possibly being the most notable example of such adaptation.

Automotive radars enabled the emergence of comfort and safety features like autonomous cruise control systems or collision avoidance systems. While fulfilling their primary role in luxury cars, they struggled to offer increasing performance, resolution and affordability. Since the market success of any new technology is usually dependent on its affordability and the car industry is not an exception here, computer vision started to play an important role in modern cars. The cameras used in computer vision are passive sensing systems and

thus do not need any complicated electronics which would drive the cost upwards. Camera-based computer vision has found important applications in systems for Lane Departure Warning or Autonomous Emergency Braking. The constantly increasing sensitivity of the image sensors can often eliminate most of the downsides of a passive sensing system. Also, some safety features like Lane departure warning simply cannot be solved with a radar-based based solution due to technical constraints (like the inability to perceive colors). The above listed reasons enabled the advancement of camera based computer vision based ADAS solutions.

With affordable sensing technologies the focus of research is being targeted towards harvesting the most out of these technologies. This means that the current research tries to precisely understand all the possible sensory inputs and attempts to develop the best safety algorithms possible that cold deal with the input data accordingly. Bringing a product to the market also involves a rigorous testing.

1.1 Thesis goal

The goal of this thesis is to evaluate the accuracy of stereo vision based object detection and ranging in context to its usage within the automotive field. A particular type of computer vision (the stereo vision) was chosen because it has some unique features that make it very attractive for the desired application as discussed in the following chapters. The real time performance of the system will not be directly evaluated and possible speed optimizations will be discussed.

2 Common obstacle detection and ranging methods

Collision (obstacle) avoidance is one of the most basic tasks for every mobile robot, which needs to navigate in a dynamic environment. To avoid collisions, the robot needs to perceive its surroundings and either stop before an obstacle or choose an obstacle-free path. Such perception can be solved in numerous ways, depending on the demands (resolution, vision speed, sensor cost etc.). The ability of environment sensing is the key to successful obstacle detection and the properties of the sensors used directly affect the final level of obstacle detection.

As said above, the level of obstacle detection depends directly on the robot's ability to perceive its environment. With simple sensors that only have a limited range and resolution, the robot's ability to "see" and possibly to react to a detected obstacle will be rather limited. The simplest sensors like ultrasonic and infrared sensors usually suffice for a slowly moving robot only, but they are affordable and thus are good to start a research with. For a higher level of obstacle detection and avoidance, the robot should perceive its surroundings in greater range and resolution, ideally creating a dense depth map of its environment.

This thesis focuses mainly on the higher level environment sensing that is enabled by advanced laser scanners (LIDARs) and stereo computer vision systems.

2.1 Sensor types

The various sensor types for obstacle detection can be divided into simpler "ranging" sensors that usually only measure distance to a single point and into more sophisticated sensors that are able to return a denser depth map of the environment.

The sensors can also be divided into active and passive systems. The advantage of all active ranging techniques (be it ultrasound, radar or LIDAR) is that they do not rely on the scene to emit energy. In contrary, passive techniques like camera vision do rely on the scene to emit or reflect electromagnetic waves. Should the scene have a limited ability to do so (e.g. at night) then an imaging sensor may have problems to detect anything at all due to its limited sensitivity.

2.1.1 Sensor types for simple ranging

The simplest ranging sensors are usually only able to simply measure range by emitting a signal in a single direction and thus are only measuring a single point in the space. This means that the resolution of these sensors is very limited. Simple ranging can be carried out by several techniques, for example with the help of ultrasonic sensing, infrared or a simple 1D laser (also called LIDAR) range sensors. The ranging may further be degraded by the width of the sensing beam which increases range uncertainty and can cause problems with ultrasonic sensors or simple infrared sensors.[1]

The limited angular resolution of 1D sensors can be worked around by combining multiple sensors into an array – then the overall resolution depends directly on the number of sensors used. However, a multi-sensor setup may quickly become difficult to mount and calibrate and also increases the cost of the whole vision system linearly according to number of sensors used.

Another way of improving these simple sensors is to rotate them around so that they scan the environment at a wider angle. This solution rather increases the sensor coverage than its resolution since the measurements still remain 1D at a time. However, for a slowly moving robot such a solution may be sufficient. The disadvantage of the rotating sensor solution is the need of a custom scanning mechanism that may be prone to mechanical failures.

The advantage of these sensors is their technological simplicity and especially their low price. Some of the main sensor types are described further.

2.1.1.1 Ultrasound

Ultrasonic distance measuring has a limited range, usually only covering several meters of distance. Besides resolution and range, the applications of ultrasonic sensing are also limited by the fact that sound propagates relatively slowly at only about 340m/s, which may not be sufficient for measurements at higher speeds. Further ranging uncertainty is introduced due to the width of the sensing beam.[1]

The advantage of ultrasonic ranging boils down mostly to the low cost of the sensors and therefore it is mostly popular in the low-cost applications and hobbyist robotics market. Ultrasonic sensors are used in production ADAS systems where there is no need for a precise and fast measurement. Example applications include parking assistants or detection of big objects in driver's blind-spots.[2]

2.1.1.2 Infrared

The infrared sensors are similar to the ultrasonic ones. They provide faster measurement due to the employment of light. Their range is usually also limited to several meters and they suffer from ranging uncertainty due to the width of the sensing beam.

2.1.1.3 Lidar

A LIDAR range sensor uses a laser beam (usually not in the visible light wavelengths but rather in the invisible /infrared/ spectrum due to eye safety reasons) which makes the measurements fast enough. LIDARs do not suffer from the beam width problems since the laser beam is very thin. However, the cost of even a 1D Lidar sensor is higher in comparison to infrared sensors.

Simple LIDAR sensing has been brought to a production ADAS system. An example of such system may be the Continental Short Range Lidar SRL 1 sensor which is capable of

Autonomous Emergency Braking functionality. The sensor uses 3 infrared beams and has a range of about 10 meters which limits its practical ADAS usage for low speeds.[3]

2.1.1.4 Radar

Simple radars with one or multiple beams have been used for some ADAS applications like Autonomous Cruise Control in production cars. Automotive radars usually have a long range up to about 200 m but with limited number of beams they too suffer from the same limitations as other 1D sensing techniques. A big advantage of radar technology is its very good resilience to various weather conditions (rain, snow, fog etc.).[2]

2.1.2 Sensor types for advanced perception

The advanced perception technologies try to sense the environment into a greater detail than 1D sensor, trying to create a dense map of the sensor's field of view. The technologies mostly employ electromagnetic waves in various wavelengths, ranging from radars, over LIDARs, to visual cameras.

The wavelengths used have a direct impact on the sensor performance with the longer wavelengths (radars) having less angular resolution.

2.1.2.1 Radar

Historically, the radars were pure 1D sensors that were only being able to measure the distance to a single point. Later, the antennas were rotated for a wider coverage, but in the world of robotics any moving part may be prone to failure. Recently, solid state automotive radars with a complex set of patch antennas are able to use dynamic beamforming for both wider angle of view and better sensing resolution. Due to beamforming ability, these solid-state radars are not limited to one dimension and can scan the scene in 2 dimensions like a common 2D image sensor.[4]

However, the resolution of such solid state radars still goes back to the wavelengths used and usually struggles to provide an acceptable lateral resolution for more precise ADAS applications like pedestrian detection. The cost of solid state radars today is still high - in the range of thousands of dollars but mass production may make them more affordable.[2]

2.1.2.2 Lidar scanners

Advanced laser scanners use one or multiple beams that rotate around a common axis. The number of beams directly influences the resolution of the LIDAR in the axial direction.

The simplest LIDAR scanners use a single rotating laser beam and thus create a cross-section image of their environment. High-end LIDARs use multiple (up to 64) vertically aligned laser beams and are able to return much denser depth information.

Advantages and disadvantages

The laser based point clouds have both advantages and disadvantages. The main advantage of LIDARs is a long range that may span up to 100 meters. Also, the precision of range

measurements does usually not degrade significantly with distance. The main disadvantage may be that the angular resolution of the scans is usually limited and depends on the construction of the laser scanner. The best scanners are able to rotate 360 degrees horizontally and have multiple laser beams arranged vertically on their spinning head. However, the vertical angular resolution even of the best laser scanners is still limited. The very high cost of the best laser scanners can also be considered a significant disadvantage. The mechanical complexity of the scanners which includes rotational parts can also be considered as a disadvantage, since these parts may fail during operation.

Another disadvantage of the laser scanners comes from their spinning nature and narrow vertical beams. Since the frequency of sensor rotation is only about 20 Hz, one full 360 degree-rotation takes about 50 ms – this may be an issue considering higher velocities of the ego-vehicle or fast lateral moving objects whose image can get deformed. Time-of-flight (TOF) cameras do remove the disadvantage of rotational scanning movement but currently at the price of wider market unavailability and very high cost.[2]

2.1.2.3 Time-of-flight camera

A time-of-flight (TOF) camera is another type of laser ranging device. The camera illuminates the scene with a short light pulse and then it measures the time-of-flight of the returning light. The camera remedies some of the disadvantages of the more common laser scanners – it is a solid-state device and it has some other positive camera-like features.

The technology was recently brought to the market of consumer electronics by Microsoft with its Kinect One sensor and by Intel with its RealSense technology. However, both of these products have a very limited range of several meters and are capable of working mostly indoors only. TOF cameras with a greater range and the ability to operate outdoors are still in the early stages of development. The main advantage of TOF camera is a better angular resolution in comparison to traditional LIDAR scanners since the camera uses a 2D photosensitive array with higher pixel density. However these cameras mainly suffer from limited range since they tend to use a common light source (introducing high light dispersion compared to focused individual laser beams) whose power has to be limited because of regulatory constraints.[5]

2.1.2.4 Digital cameras

Digital cameras can also be used to gain depth information of the scene. There are two basic types of systems – monocular cameras and stereo cameras.

2.1.2.4.1 Monocular camera

A single (monocular) camera tries to semantically understand the scene from a single image. Multiple cameras may be used to extend coverage, but not to gain additional information about the same scene. The advantage of monocular systems is their hardware simplicity. However, the depth perception ability of monocular vision is considerably limited. In order to be able to gain depth from monocular vision, prerequisites like a known

scene geometry (flat surface) or camera motion (structure from motion) are needed, which complicate the practical applications.

2.1.2.4.2 Stereo camera

In the case of stereo vision two or more cameras can be used to gain depth from the scene by finding correspondences between static camera images and then calculating the depth using triangulation. The cameras capture the same scene but do have an offset (vertical or horizontal). Due to a slightly different viewing angle of both cameras, the same point in the scene appears at different locations in the left and right images. The distance between these pixels and camera parameters are used in triangulation to estimate depth. The algorithmic problem in stereo vision is finding correspondences for all the pixels between both images.

Stereo computer vision does not try to semantically understand the scene. The goal is to find correspondences between the two images to be able to use triangulation to extract a 3D model of the captured scene. With the semantic part missing, the algorithmic task is clearly defined, thus stereo vision tends to be algorithmically simpler than monocular vision.

Advantages and disadvantages

The use of stereo vision for 3D perception has both advantages and disadvantages. The low cost of the image sensors and their high resolution, which results in a denser point cloud, can be considered as the main advantages. Another advantage is that a stereo camera is a solid-state device without any moving part and thus is not prone to mechanical failures.

The main disadvantages of a stereo camera are the need for a textured scene for a good function of stereo matching algorithms - the rather complex stereo matching methods mostly do not produce a noise-free disparity estimation and also may require a lot of computational power. Furthermore, the depth precision depends on physical camera factors like pixel density of the image sensor, focal length and camera baseline. The challenges of the correspondence search are described in the stereo matching section below.[6]

2.2 Obstacle detection

2.2.1 Simple obstacle detection

The simplest systems directly map the result of distance ranging to robot steering control and the collision avoidance is done via simple ranging conditions – for example if the measured 1D distance is lower that a given threshold (and thus it is probable an obstacle is in the way), the system tells the robot to turn right or left and is hoping that it will find a way without obstacles (where the measured range will be long enough). In the case of simple 1D ranging sensors, the robot has no idea what the scene on the right or left from the sensor beam looks like. When multiple 1D ranging sensors are used, the robot may have a better idea of its surroundings with some more angular resolution, but the basic

principle of operation remains the same. The angular resolution is very low even with multiple 1D sensors.

2.2.2 Advanced obstacle detection

Higher-level obstacle avoidance systems usually consist of two parts – the vision system which provides a 3D point cloud of the environment and the processing part which tries to analyze the point cloud and possibly a take steering or braking action based on its findings.

The most widely used sensor for obstacle detection across various industries today is the laser scanner. Even though it may produce rather sparse depth information the main advantages are the precise measurements and its setup simplicity. The higher sensor cost may get amortized quickly for an industrial application. The most used scanning scenario are simpler applications that only require a small range and a single scanning beam is sufficient. Such scenarios may involve mobile industrial robots or autonomous vehicles that operate at slow speeds in limited environments.

2.2.2.1 Obstacle detection from mono camera

Obstacle detection from a monocular camera may be solved as an inverted problem by the estimation of freely drivable space. Since no real 3D information can be extracted from a single camera image (except when in motion), for a common traffic situation it can be assumed that the terrain ahead is flat and that the camera height above ground and its pitch angle is known. If these conditions are met it is possible to extract the distance to a potential obstacle.

The estimation of drivable collision-free space from monocular video was presented by Yao and others. The team reduced the free space estimation task to an inference problem on a 1D graph, where each node represents a column in the image and its label denotes a position that separates the free space from the obstacles. The algorithm exploits several image and geometric features based on edges, color, and homography to define potential functions on the 1D graph, whose parameters are learned through a machine learning method called structured support vector machine (SVM).

The free space estimation was tested on two different datasets containing road and water scenes. However, it was found that the results of the algorithm degrade when there are road markings or shadows in the image.[7] Further academic research was also conducted on the topic of drivable area estimation from monocular vision.[8]

Another free driving space estimation with monocular vision has been done by the Mobileye company which is a supplier of vision-based ADAS systems for multiple car manufacturers. While there is no detailed information available about the algorithms used in the system it can be assumed that the company's production systems are using it.[9]

2.2.2.2 Obstacle detection on a laser point cloud

Since the 3D sensing technologies use very different hardware to acquire depth information, the resulting point clouds also look differently. For instance, the Velodyne HDL-64E laser scanner uses 64 vertically aligned laser diodes to scan the environment around the vehicle in a rotational motion, so the final point cloud that depicts a flat surface consists of multiple centered "rings" with various diameters. The Velodyne LIDAR has a horizontal angular resolution of 0.08 degrees, which makes about 4500 points per beam on 360 degrees. By counting all 64 beams, about 288 thousand points are detected with a single sensor rotation. However, the vertical angular resolution is still limited and may not be sufficient to detect all kinds of obstacles. Special algorithms that take the physical construction of the LIDAR into account are needed.

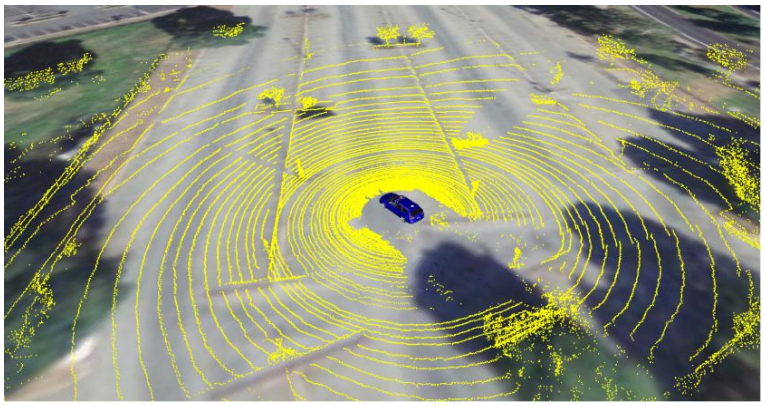

Figure 1: A typical result of Velodyne LIDAR scanning (in yellow).[10]

2.2.2.2.1 Stanley 2005

One of the projects that dealt with obstacle detection on laser point cloud is the Stanley robot developed by Stanford Racing Team for the DARPA Grand Challenge in 2005. Stanley was based on Volkswagen Touareg and was equipped with five SICK single-beam laser scanning range finders mounted on the roof, tilted downward at various angles to scan the road ahead. In Stanley's software, the obstacle detection was solved as a classification problem assigning to each 2D location in a surface grid one of three possible values: occupied, free, and unknown. A location was marked as occupied by an obstacle if Stanley can find two nearby points whose vertical distance exceeds a critical vertical distance. The classification resulted in a 2D map of obstacles and free driving space.

However, applying such classification solely based on laser data was found to be insufficient for reliable robot navigation, because a small error in the vehicle's roll/pitch estimation led to a massive terrain classification error. The vehicle pose errors were magnified because the lasers were aimed at the road up to 30 m in front of the vehicle. On a reference dataset of labeled terrain, it was found that even for roll/pitch errors smaller than 0.5° more than 12% of known drivable area was classified as obstacle for a height threshold parameter of 15 cm.

It was found that the errors in obstacle classification were due to part of the terrain being scanned multiple times by various lasers and that the errors were strongly correlated with the elapsed time between the two scans. Hence, a probabilistic error testing was implemented to eliminate the pose estimation error. Further data-driven parameter tuning was also performed via managed driving over obstacle-free terrain and a learning algorithm was developed.

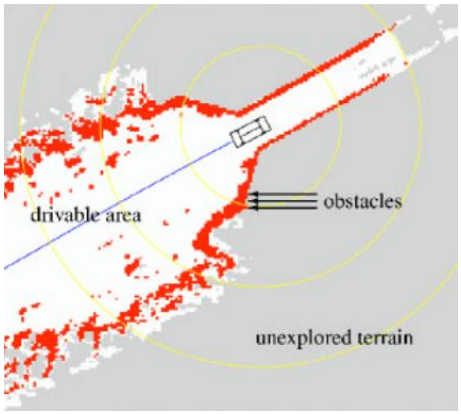

Figure 2: Terrain map produced by the Stanley vehicle.[11]

The range of laser obstacle detection was about 22 meters, which was sufficient for obstacle avoidance in speeds up to 25 mph. However, since the robot needed to travel at the speed of 35 mph, the final robot design was equipped with a supplementary camera-based vision system that had a greater range of more than 70 meters.[11]

2.2.2.2.2 Junior 2007

In 2007, the Stanford Racing Team entered the DARPA Urban Challenge competition with another robot called Junior, which was based on a Volkswagen Passat B6. In comparison to the Stanley robot, the laser ranging sensors were upgraded to a more sophisticated Velodyne HDL-64E laser scanner with 64 laser beams. Like with Stanley, the laser based obstacle detection needed some custom approach. At first, a simple algorithm was developed that finds points with similar ground coordinates whose vertical displacement exceeds a given threshold. However, due to sensor's range and calibration error, the algorithm was only able to detect large obstacles such as pedestrians, signposts, and cars.

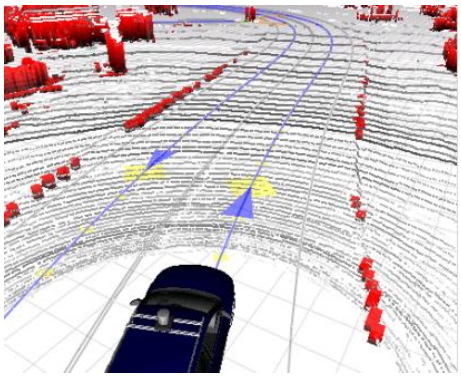

Figure 3: Map of the environment generated by the Junior vehicle.[10]

Further development was needed to enable the detection of smaller obstacles. As the LIDAR sensor rotates, the beams sweeps out in circles of fixed radius on a flat terrain. The inter-ring distance of circles produced by adjacent beams becomes lower with an angled obstacle or sloped terrain. Thus, obstacles that are not apparent to the vertical thresholding algorithm can be reliably detected by finding points that generate inter-ring distances that differ from the expected distance by more than a given threshold.

The above described algorithm is sensitive to vehicle's rolling and pitching which causes the "laser rings" to compress and expand. A workaround to this problem is making the expected distance to the next ring a function of range, rather than the index of the particular laser. Like this, as the vehicle rolls to the left, the expected range difference for a specific beam decreases as the ring moves closer to the vehicle.[10]

2.2.2.2.3 Tatran Racing (DARPA Urban)

The winning team of DARPA 2007 event was the Tatran Racing team from Carnegie Mellon University. Their obstacle detection system worked by computing a cost map representing the "traversability" of the terrain. The algorithm relied on comparisons of pairs of laser points, which calculated a cost based on the elevation difference between the two points and the angle of the vector connecting them, relative to the ground. The system maintained a history of points as the vehicle traveled to ensure that obstacles of a certain height are seen. This ensured that the downward looking lasers scanned a constant vertical distance for each planar location.

It was found that the accuracy of obstacle placement is clearly dependent on the sensor calibration and the resolution of the cost map. Insufficient sensor calibration may improperly transform laser data to world space. It was found that the algorithm properly placed cones within 25 cm of their location which was in most cases sufficient to navigate through obstacle fields with cones and walls.[12]

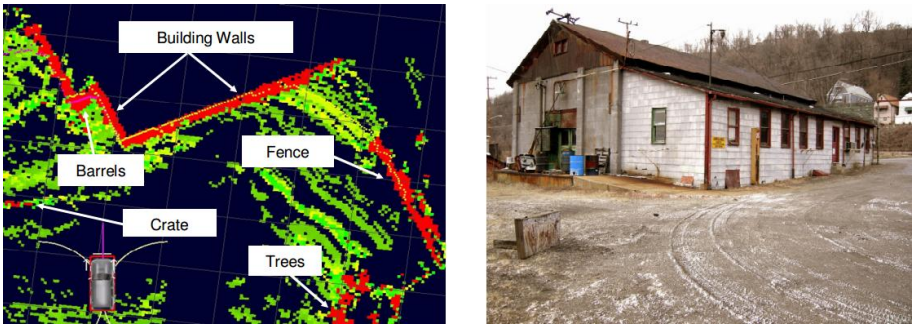

Figure 4: Map of the environment generated by the Tatran Racing vehicle.[12]

2.2.2.2.4 Lidar based obstacle detection conclusions

The DARPA competition robot examples above demonstrate that due to the limited vertical angular resolution of the laser scanners the detection of smaller obstacles is a challenge that has to be solved by additional means. Building a ground obstacle map with

consecutive laser measurements was found to be prone to roll/pitch estimation errors which are expensive to solve in hardware and a rather difficult software solution to this problem had to be developed.

2.2.2.3 Obstacle detection with stereo camera

There are numerous projects that research stereo vision for its potential applications. This thesis focuses on stereo vision applications for collision avoidance in road transportation. One of the leading research projects that deal with usage of stereo vision in passenger cars is the Italian company VisLab.

2.2.2.3.1 TerraMax

The only one of the 5 vehicles that finished the DARPA Grand Challenge in 2005 that has a stereo vision based obstacle detection system was the TerraMax truck developed by Oshkosh Truck Corp, University of Parma and other partners. The truck was equipped with three forward-looking cameras that formed a stereo camera system that had variable baseline functionality. Since the disparity is inversely proportional to depth, a system with a wider baseline has greater depth precision for longer measurements.

Figure 5: The TerraMax vehicle.[13]

The baseline between the two outmost cameras was about 1.5 meters and the central camera was placed asymmetrical at about 0.5 meters from the right one. The system therefore consisted of 3 separate stereo cameras with the baselines of 1.5, 1.0 and 0.5 meters. Various baselines were used for obstacle detection at various speeds of the truck.

Image disparity was first used to estimate the average terrain slope in front of the truck. Slope information was then used for obstacle detection, while any significant deviation from the average smooth slope detected previously was identified as an obstacle. The exact location of obstacles was then obtained via stereo triangulation. Even though a fairly precise localization was obtained, the measurement was further refined with data from a laser scanner, which was placed into the front bumper. This enabled the system to detect thin vertical posts and fence poles.[14]

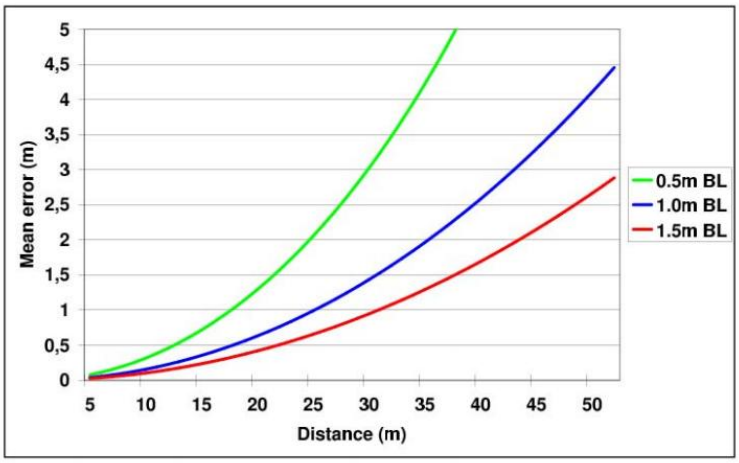

Figure 6: TerraMax - dependency of the stereo camera depth error on distance.[13]

The stereo vison system ran at 15 frames per second and proved very robust against false positives. Only two cameras (one stereo camera) were active at a time and the cameras were synchronized through the FireWire bus. The similarity measurements on a stereo pair ware made via a SAD (Sum of Absolute Differences) algorithm which allowed code optimizations to use processors MMX and SSE instruction sets.[13]

2.2.2.3.2 VisLab

The VisLab company sprung off the University of Parma and develops stereo vision systems. One of the company's projects is the development of an autonomous car. The sensing of their autonomous car is heavily based on stereo vision and minimizes the usage of other sensing types like ultrasound, radars or LIDARs. Its latest autonomous car prototype called DEEVA is equipped with multiple stereo camera systems that together cover 360 degrees horizontally around the vehicle and thus can substitute much more expensive LIDAR scanning systems.[15]

2.2.2.3.3 Daimler

Researchers at Daimler AG - Research & Development have been working on a vision based ASAS system since the 1990s. A group of scientists around Uwe Franke developed a stereo-vision based collision avoidance system that found its way to Mercedes Benz production cars in 2013 (S and E Class) and 2014 (C Class). The system is based on a research project called "6D-Vision" – it perceives the environment in front of the vehicle and is able estimate both position coordinates and velocity vector of other traffic participants like cars, bicyclists or pedestrians. The system is able to predict the position of moving objects 0.5s in advance even before they cross the driving corridor. The system has been implemented onto an FPGA.

6D-Vision uses stereo vision to determine the 3D position of image points. In addition to that optical flow is used to track an image point over several images in a sequence. Since each point has a known 3D position, 3D-velocity of the point can be determined with

suitable filtering. The combined 3D position and 3D velocity yield the six dimensions of the method.

The image sequence in Figure 1 shows a collision risk scenario, since the kid will be in 1 second at the same position as the vehicle where the cameras are mounted. In order to detect this, the Computer has to determine both the position and speed of the kid.

Figure 7: The 6D-Vision algorithm detecting a pedestrian behind a vehicle.[16]

Technical details

The stereo matching algorithm used in the stereo vision system is "Semi-Global Matching" (SGM). To determine depth, this algorithm performs an optimization step in which it exploits couplings between neighboring image points. This algorithm was further optimized and developed to maintain a high performance level also at night time and in bad weather conditions. The first real-time implementation of SGM on a low-power and inexpensive FPGA (Field Programmable Gate Array) was performed in 2008 and was able to compute stereo images 25 times per second. This implementation has a detection range of up to 50m.

The object velocity estimation is done using optical flow which determines the temporal displacements of image points in an image sequence. The typical optical flow methods are only able to estimate small displacements. However, in a common traffic scene, especially in areas close to the vehicle or in curves, large displacements may occur. In 2004 Daimler developed an algorithm that solves this problem efficiently computing the optical flow using the census transform. Arbitrary large displacements are computed with a constant and small computation time and reliable results are obtained even when driving fast or under adverse weather conditions.

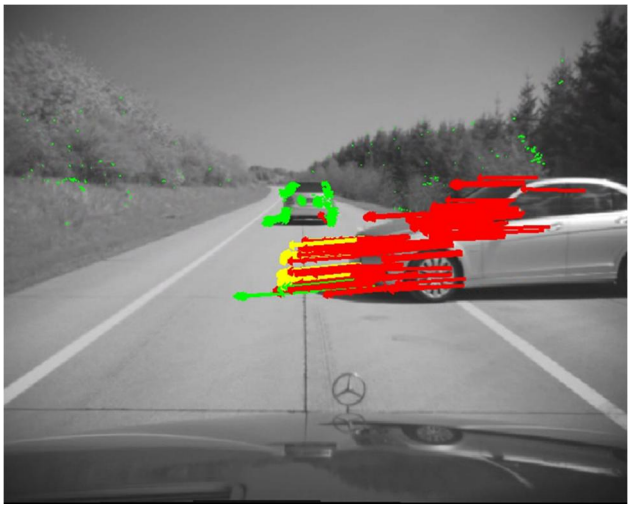

Figure 8: 6D-Vision detecting a vehicle entering the driving corridor from the right side.[16]

6D-Vision is presented to be a "track before detect" algorithm. This means that a motion is determined on an image point level before an object is formed by grouping. This method avoids the typical errors of other stereo methods that first form objects and then try to determine their motion by tracking the object. There are situations where the advantage of point level tracking becomes apparent – those situations are especially partially occluded objects, objects that are close together in space or difficult viewing conditions. With the 6D information, the collision risk can be determined for every single point of the image and independent of an object detection step (see Figure 8). Red arrows depict imminent collision risk, yellow a close encounter, and green no collision risk. In the scenario in Figure 8 an automated emergency brake was initiated to avoid a collision.

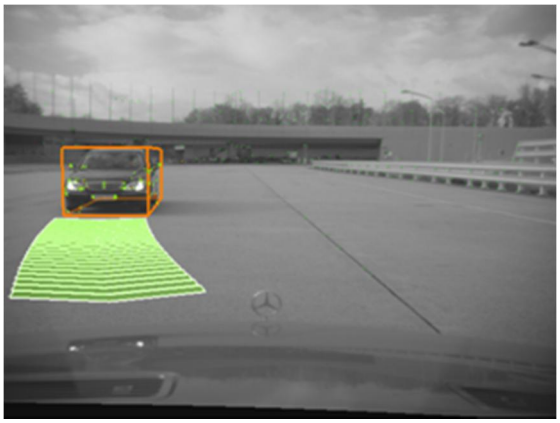

Figure 9:6D-Vision estimating the future path of an oncoming vehicle.[16]

6D-Vision detects objects by grouping compatible motion vectors that are close to each other. The motion itself is determined by averaging the 6D motion information. For vehicles and cyclists, the measurements are combined in an additional Kalman filter to determine the rotational speed besides position, size and speed. This is necessary to obtain accurate predictions 1-2s in advance in order to distinguish between collision-critical and uncritical traffic situations. Figure 9 shows that the driver of the oncoming car is about to

make a turn into the path of the ego-vehicle. The 6D-Vision system needs only about 200 milliseconds to detect potential collisions for the full field-of-view of the camera.[16]

2.2.2.3.4 Subaru

Subaru is the automobile manufacturing division of Fuji Heavy Industries (FHI) conglomerate. Together with Hitachi, FHI developed a stereo-camera based ADAS system that found its way to a production Subaru Legacy model, which went on sale in 2008. The system has a collision avoidance function together with other features like lane departure & sway warning, adaptive cruise control and pre-collision throttle management.[17][18] The Outback and Legacy models equipped with the system ranked at the top position in the first ever IIHS crash avoidance system ratings in 2013 at the speeds of both 12 and 25 mph.[19]

2.2.2.3.5 ADAS tests

Thatcham Research conducted a study on Autonomous Emergency Braking systems. 11 vehicles were present in the test and were equipped with various AEB technologies like radar, LIDAR and cameras. The best performer in the test was the Subaru Outback, which was equipped with a stereo camera based AEB system. The Outback was the only vehicle able to achieve the highest performance level from the sample tested, showing full collision avoidance at 50km/h.[20]

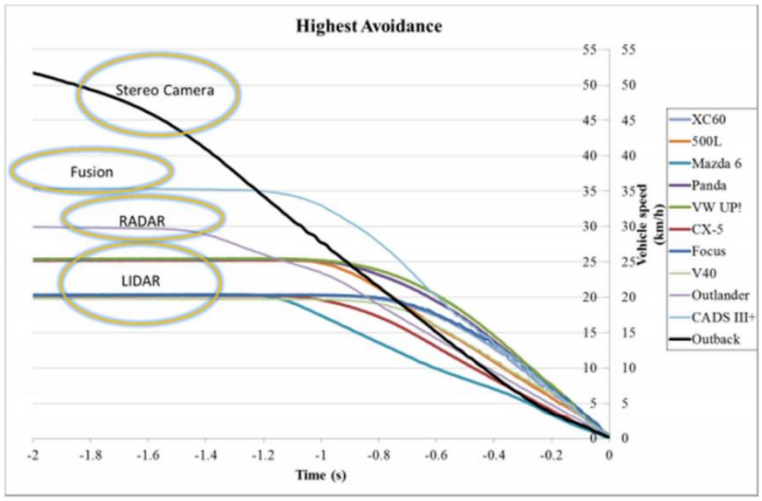

Figure 10: Time history for the highest avoidance speed for each tested vehicle with marked sensor types.[20]

2.3 Stereo vision

This thesis deals with stereo vision object detection and ranging therefore a special chapter is included that introduces the basic stereo vision principles.

2.3.1 Basic principles

A stereo camera system consists of two cameras which have the following features:

- image planes of cameras are parallel to each other and to the baseline
- camera centers are at same height

- focal lengths are the same
- camera lenses are free from distortion

If the above conditions are met, then epipolar lines fall along the horizontal scan lines of the images (see Figure 11). If at least one of the conditions is not met then a camera calibration is needed which introduces correction mechanisms to bring a stereo system to a desired state.

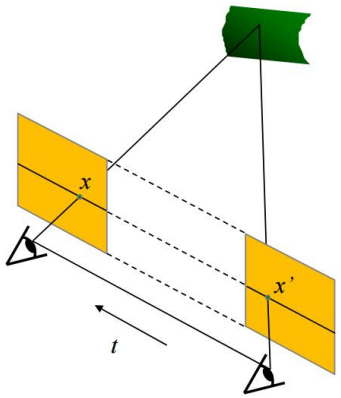

Figure 11: Two parallel image planes showing a common epipolar line.[21]

Stereo image rectification is needed if the camera image planes are not parallel to each other. During this step images planes are reprojected onto a common plane parallel to the line between optical centers.

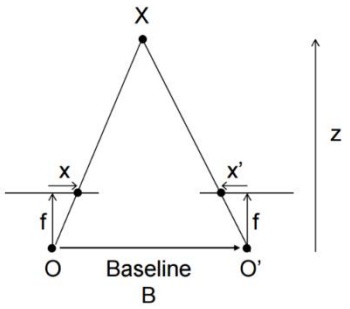

Figure 12: Depth computation from image disparity.[21]

The distance from the optical center of the stereo camera can be calculated due to the triangle similarities of the stereo setup as shown in Figure 12.[21]

2.3.2 Stereo matching

The biggest algorithmic problem of stereo vision is finding the correspondences between the left and right image. Thanks to image rectification, the correspondence search is limited to a single dimension only - the search direction is parallel to the camera baseline.

The source pixel or block of pixels from one image is usually being looked for in the other image along the epipolar line.

There may be numerous problems during stereo matching - since both images are taken from a different angle, occlusions can occur. Some of the stereo matching algorithms try to take these effects into account, usually at a higher computational cost, which may not always be appropriate.

Stereo vision/matching challenges:[6]

- occlusions and discontinuities
- texture-less regions
- repetitive patterns
- specular surfaces
- transparent objects
- photometric distortions and noise
- foreshortening
- perspective distortions

There are multiple stereo matching algorithms, ranging from the simplest and computationally inexpensive to more sophisticated and computationally expensive. Some of the simplest algorithms evaluate the disparity through a matching cost. Some of these algorithms include Cross-correlation, Sum of Squared Differences (SSD) and Sum of Absolute Differences (SAD). Usually a square matching window is used for the evaluation. Most of the methods evaluate only pixel brightness, so images are converted to grayscale before processing.[22]

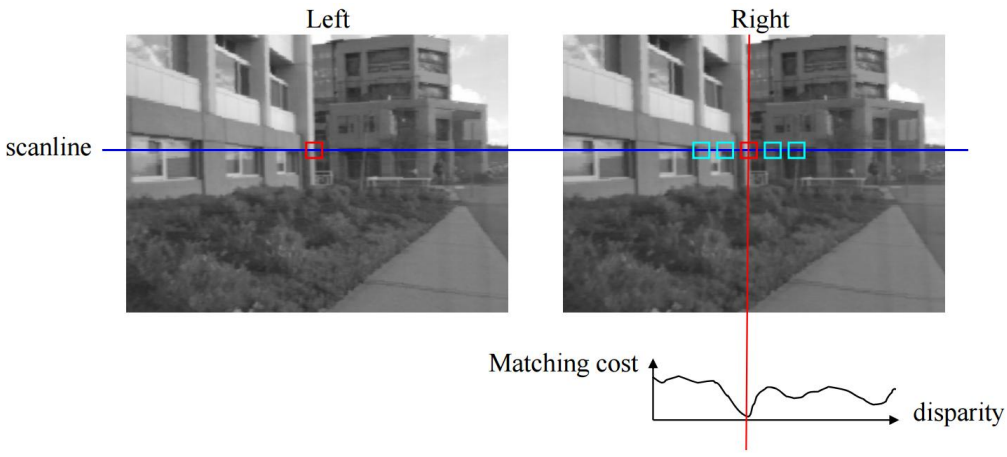

Figure 13: Correspondence search with similarity constraint. The cost function shows the matching cost for a single source window in the left image.[21]

Two stereo matching algorithms will be discussed in greater detail here – Block Matching (BM) and Semi-Global Matching (SGM). These algorithms are being dealt with further in

the object detection algorithm due to the fact that they are implemented within the OpenCV open source library.

2.3.2.1.1 Block Matching

One of the simplest stereo matching algorithms is called block matching. Block matching usually employs the Sum of Absolute Differences (SAD) method to measure the similarity between image blocks. The blocks are usually squares whose sides have an odd number of pixels (9x9, 15x15 etc.) to allow proper centering of the evaluated pixel. The block around the evaluated pixel in first image works as a template – a matrix is created from its pixels' brightness values. In the SAD method another blocks' brightness matrices along the epipolar line in the second image are subtracted from the template and the sum of the absolute values of the resulting matrix is saved. The block of the second image which has the smallest absolute difference to the template in the first image is the most similar to the template and thus is marked as a "stereo correspondence".

The Sum of Absolute Differences is computed in following way:

$$\sum_{(x,y)\in W} | I_R(x,y) - I_L(x+d,y) |$$

where I_R and I_L are the pixel intensity values of the right and left image at (x, y) and the d is the disparity.[22]

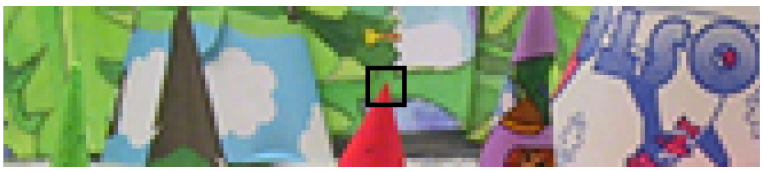

Figure 14: SAD detection window in the left image.[23]

Figure 15: SAD detection window in the right image.[23]

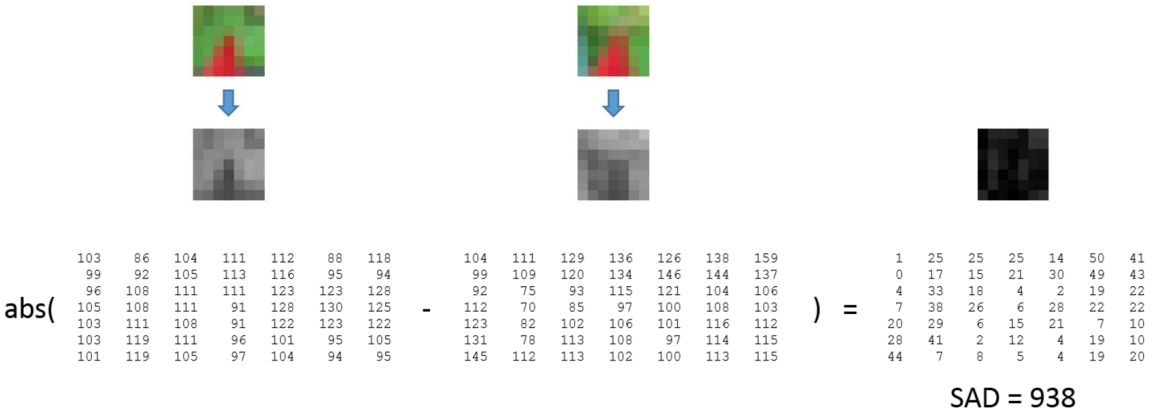

Figure 16: Sample intensity matrices for left and right images and the final SAD matrix.[23]

2.3.2.1.2 Semi-Global Matching

The Semi-Global Matching (SGM) algorithm was developed by Heiko Hirschmuller in the German Aerospace Center (DLR). The method tries to find correspondences for every pixel and is supported by a global cost function, which is optimized in 8 path directions across the image. An advantage also is that the method has a regular algorithmic structure and uses simple operations, which enables the possibility of parallel processing implementations on various hardware platforms. Even though SGM is not the winning method in international benchmarks that only evaluate quality (like the Middlebury benchmark or The KITTI Vision Benchmark Suite), it offers a good combination of speed, quality and robustness which are valuable features for a vision system.

Daimler Research uses SGM in their driver assistance system research project called 6D-Vision. This project was applied in a production ADAS application that started to be offered as an option for several Mercedes-Benz passenger car models since 2013.[24][25]

2.3.3 Stereo vision depth accuracy

The depth accuracy of a stereo camera is expressed with following equation:

$$dz = (z^2 * de) / (f * b),$$

where dz is the depth error in meters, z is the depth in meters, de is the disparity error in pixels, f is the focal length of the camera in pixels and b is the camera baseline in meters. It can be seen that if the disparity error is constant the depth error grows with the square of distance for a given stereo camera system. If the disparity error is constant the depth error in given distance z can be lowered by either wider stereo camera baseline or larger focal value.[26]

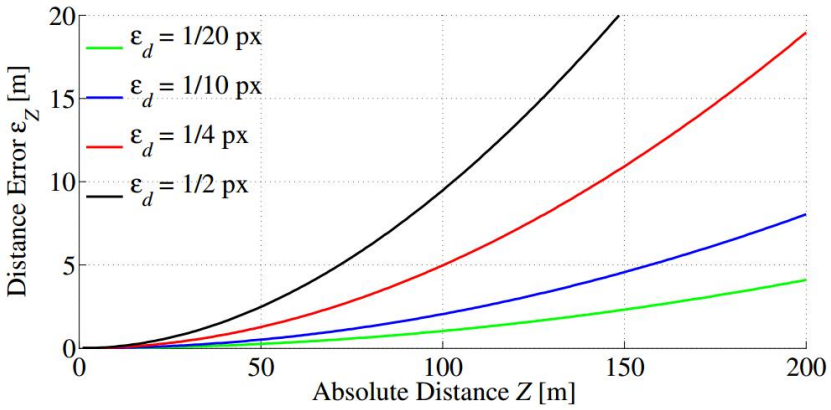

Figure 17: Metric distance errors ε_Z increase quadraticaly for given stereo disparity errors ε_d for a reference stereo system with given baseline and focal length.[27]

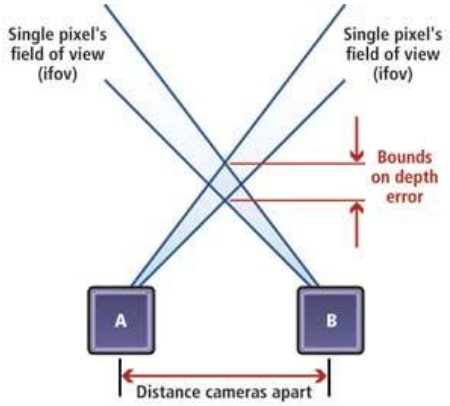

Figure 18: Image depicting the area of uncertainty in depth estimation using stereo camera.[28]

3 Object detection and ranging

Stereo vision can provide an inexpensive yet quite precise 3D sensing system. An object detection algorithm that can handle noisy point cloud input data is proposed and its ranging capabilities are evaluated on three traffic scene sequences. The object detection together with ranging can find various applications in automotive fields or in robotics.

The detection algorithm can be parametrized to allow for tuning according the expected object size and point density. A detection window with variable size is applied and a point density coefficient can be set.

3.1 Generating point cloud

The point cloud is generated in two steps. At first, a stereo matching algorithm produces a disparity image of the scene by finding corresponding pixels in the rectified stereo image pair. In the next step, each pixel of the disparity image is reprojected to 3D space with a set of simple equations. The reprojection is performed for each dimension separately with a single equation resulting in a set of 3 equations:

$X = (x-cx)*base/d$

$Y = (y-cy)*base/d$

$Z = f*base/d,$

where (x,y) is a 2D point in the image coordinate system, (cx,cy) is the principal point (image center) of the stereo camera, f is the focal length of the camera, base is the baseline of the camera, d is the disparity and (X,Y,Z) is a 3D point in the camera coordinate system.[29]

Figure 19: An example reprojection from the camera point of view. No depth information is available for the black regions because the disparity was not estimated during stereo matching.

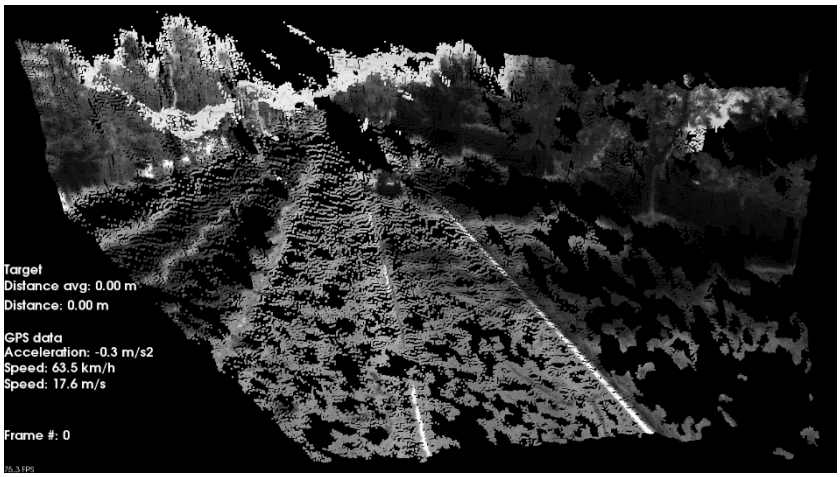

Figure 20: A point cloud seen from a different viewing angle than the camera lens.

3.2 Point cloud filtering

A point cloud that represents a 3D scan of the scene was generated in previous step. Such point cloud can be very large both in terms of total number of points and also in absolute 3D dimensions. The point cloud can also contain noise from stereo correspondence mismatches.

3.2.1 Cropping

If a subset of the point cloud (that can be defined with analytic equations) is only needed then the point cloud can be cropped easily by applying these equations to each of the 3 dimensions. The equations form a bounding box for the subset. This is useful when only particular area of the point cloud is of interest – for example the approximate driving corridor in front of the moving vehicle.

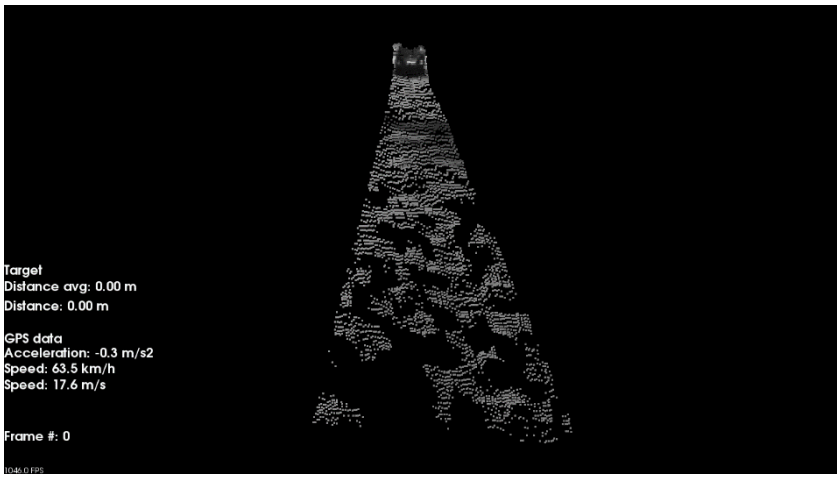

Figure 21: Point cloud after cropping to approximate vehicle driving corridor.

3.2.2 Noise filtering

The point cloud generated with stereo matching algorithm may contain points that are not valid real-world objects since they originate from stereo matching errors. Such points may be falsely detected as objects. Dedicated noise filtering in the point cloud before the

detection phase is not employed in this work. Instead a detection algorithm is proposed that is immune to most of the noise that originates from the StereoBM matching method given the detection algorithm parameters are set properly.

3.3 Object detection in point cloud

A point cloud was generated and filtered in previous steps. The closest object with given parameters will be detected in this chapter.

3.3.1 Detection algorithm

3.3.1.1 Principle

The object detection algorithm itself works on a simple principle – it is searching for points that are located close together in space and thus there is a high probability that they form an object (obstacle). The algorithm evaluates the number of neighboring points for points in the cloud which means that two loops are needed – a main outer loop and a nested inner loop. The point cloud is only sorted by Z value (distance) but otherwise unorganized which means there is a risk that if no object is found the algorithm completely traverses through both loops. The time complexity of the algorithm is therefore quadratic ($O(n^2)$).

3.3.1.2 Detailed description

The point cloud is a 1D array that is sorted by the distance of the points from the camera (from low to high). In the outer loop the algorithm iterates through all the points in the cloud and only picks points that fall into given criteria - points that are in the driving corridor of the vehicle and points that are above the ground. A point is placed in the center of a cuboid-shaped bounding box – the X and Y dimensions of the cuboid depend on the algorithm parameters and the Z dimension is a depth tolerance that is calculated automatically.

Considering a simple cuboid object detection case for straight driving – the algorithm uses a rectangular window in in the X-Y plane (XYWindow) and has a dynamic interval within the Z-axis (depth) that increases with distance because of stereo triangulation uncertainty. The detection algorithm has 3 input parameters – two metric (detection window width and height) and one coefficient that represents expected point density for detection window (decimal value ranging from 0 to 1 and meaning 0-100 %):

- windowSizeX (metric)
- windowSizeY (metric)
- minNeighbourCoef

The Z-axis interval is computed as follows:

zTolerance = distance / 10,

where distance is a distance of the evaluated point from the camera. This equation goes against the fact that the depth estimation error in stereo vision grows with the square of

actual depth (see above) but it was found experimentally that this equation approximates the depth error well enough for the vast majority of measurements.

The point cloud may contain noise in the form of invalid 3D points that come from stereo mismatches. It was found that with a reasonable big XYWindow (1x1 m was tested) such noise was not detected as an object.

The crucial part of the detection algorithm is estimating the natural point density in the evaluated distance so that the metric dimensions of the XYWindow can be converted to number of points within XYWindow in the point cloud. The natural point density estimation is described later in this thesis.

The detection is performed above a given threshold in the Y axis (height) to exclude the road surface from the algorithm. In the end a median value from the set of points in the detection object is taken and this value is considered to be the distance to the object. The median value is used because it is robust against noisy data.

3.3.1.3 Pseudocode

The point cloud is an unorganized 1D array of point objects. The array is sorted by the Z value of the points (distance from camera) before processing. This is an optimization that brings better performance.

The detection algorithm in pseudocode:

```
set zValue to 0
start main loop (all points in point cloud)
        if source point is in driving corridor (testCorridor)
                determine minNeighbourCountInDistance
                neighbourCount = 0
                start inner loop (all points in point cloud)
                        if (neighbourCount >= minNeighbourCountInDistance) break
                        if target point is close to source point (testTarget)
                                neighbourCount++
                        end if testTarget
                end inner loop
                if (neighbourCount >= minNeighbourCountInDistance)
                        zValue = source point z-value
                        break
                end if
        end if testCorridor
end main loop
return zValue
```

3.3.1.4 Point density in distance

The crucial part of the detection algorithm is estimating the natural point density in the evaluated distance so that the metric dimensions of the XYWindow can be converted to number of natural points in a given distance from camera in the point cloud. There are two

solutions of this problem. The first solution is to use the pinhole camera model based where a point is projected onto a plane using central projection. Thanks to the central projection the density of the points on the plane in given distance can be estimated. The second solution is to normalize the point cloud so that is has the same point density in its entire space. For the normalization, a voxel grid filter can be used. In this thesis the detection algorithm uses the central projection model.

3.3.1.4.1 Using pinhole camera model

The digital camera used to acquire images for stereo processing uses the pinhole camera projection model that is based on central projection. Together with rectilinear propagation of light this fact means that the density of pixels in given distance from camera lens drops with the square of this distance. This principle is known as "inverse-square law". To give an example – if a square window with dimensions of 1x1m is covered by X camera pixels in the distance L, the same window is covered with only X/4 pixels in distance 2L. The pixel density in distance from camera is influenced by the resolution of the image sensor and by the focal length of the camera lens.

With the inverse-square law in mind following assumption is made on the pixel density in given distance from the camera:

number of pixels on S in the distance L = (number of pixels on S in one unit of distance L) / L^2,

where

S is the area of the window and L is the distance from camera lens.

3.3.1.4.2 Mathematical inference

The equation above will be mathematically inferred now and the number of pixels on S in one unit of distance L will be found. The expected pixel count in the windows is called minNeighbourCountInDistance and is defined as following:

minNeighbourCountInDistance = minNeighbourCoef * windowWidthPixels * windowHeightPixels,

where

minNeighbourCoef is a real coefficient ranging from 0 to 1 (represents 0-100% pixel density),

windowWidthPixels is the width of the windows in pixels and

windowHeightPixels is the height of the windows in pixels.

The previous equation is substituted with following:

windowWidthPixels = sensorWidthPixels * windowSizeX / sceneWidth

windowHeightPixels = sensorHeightPixels * windowSizeY / sceneHeight

where

sensorWidthPixels is the horizontal pixel count of image sensor,

sensorHeightPixels is the vertical pixel count of image sensor,

windowSizeX is the window width in metric units,

windowSizeY is the window height in metric units,

sceneWidth is the scene width captured by camera in given distance and

sceneHeight is the scene height captured by camera in given distance

It is further substituted with following:

sceneWidth = 2 * distance * cameraFovHTan

sceneHeight = 2 * distance * cameraFovVTan

where

distance is the distance of evaluated point from the camera lens,

cameraFovHTan is the tangent value of half the horizontal FOV camera angle and

cameraFovVTan is the tangent value of half the vertical FOV camera angle.

The tangent values are computed in following way:

cameraFovHTan = sensorWidthPixels / (2 * cameraFPixels)

cameraFovVTan = sensorHeightPixels / (2 * cameraFPixels)

where

cameraFPixels is the focal length of the camera in pixels.

The equations above are substituted back to the original equation and the following final equation is yielded:

minNeighbourCountInDistance = (minNeighbourCoef * windowSizeX * windowSizeY * cameraFPixels^2) / distance^2,

where all of the values except distance are constants for a single image frame or single point cloud. The inference above confirms the original assumption on the number of pixels in given area to be inversely proportional to the square of distance from camera lens. It has been confirmed that the pixel density behaves according to the „inverse-square law“.

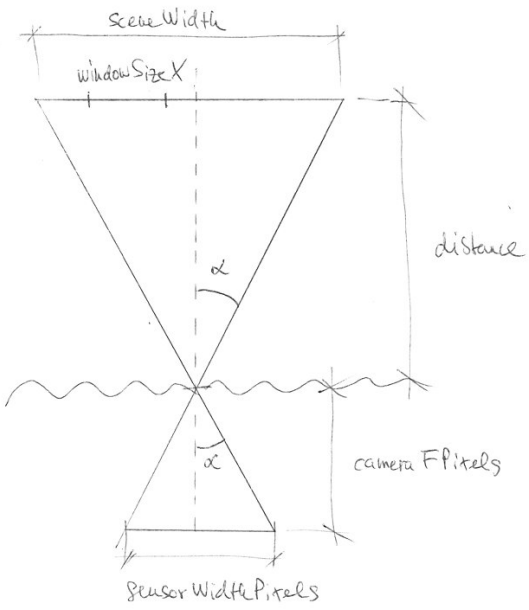

Figure 22: Diagram showing main principle of point density estimation in given distance from camera lens.

The use of pinhole camera model for object detection in point cloud has the advantage that the detection can be performed even in large distance from the camera since the pixel density is computed very precisely. However, a disadvantage of this method is that objects very close to camera lens have a very high pixel density and thus the quadratic complexity of the detection algorithm comes into play and the detection itself is not suited for a real time environment very well. This burden can be overcome by normalizing the point density within whole point cloud or by lowering the time complexity of the detection algorithm.

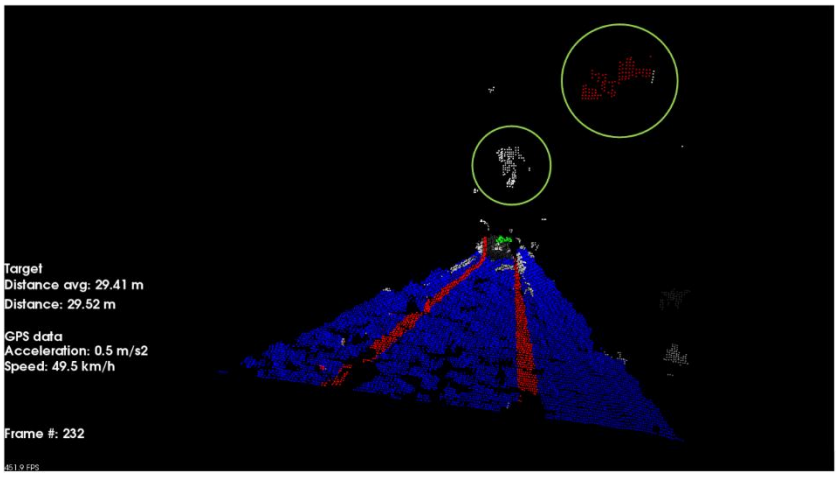

Figure 23: A leading vehicle is successfully detected in the background while the point cloud noise (circled) was correctly left out from the detection.

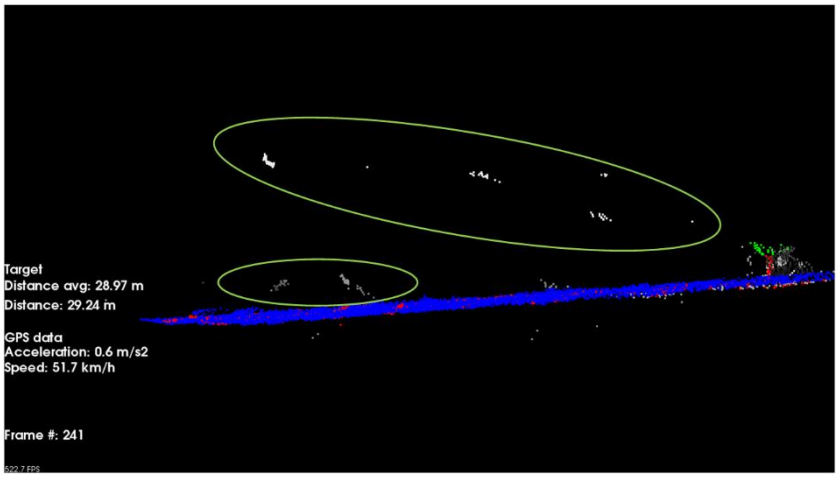

Figure 24: A leading vehicle is successfully detected on the right (side view) while the point cloud noise (circled) was correctly left out from the detection.

3.4 Real time considerations

The proposed object detection algorithm has a quadratic time complexity of $O(n^2)$ which means it is not well suited for real-time performance especially for close and large objects since such detections result in very large inputs of data. With this in mind, examinations on possible real-time optimizations had been done. The object detection speed can be optimized by either using "stronger" hardware for the computations themselves or by applying software optimizations. Some possible software optimizations are being discussed further.

Software optimizations can be performed at various phases of the computer vision "pipeline" (stereo matching, point cloud filtering or object detection). The detection step takes the longest time in the current implementation so its optimization for speed will be the most effective optimization.

None of the further proposed real time optimizations were evaluated in this thesis because it was not part of the assignment.

3.4.1 Stereo matching

3.4.1.1 Input resolution

The input resolution directly affects the number of points in the resulting point cloud. It was found that reducing the resolution to half the width and height (4 times less pixels) does not considerably influence the object detection. Even when reducing the resolution to quarter (16 times less pixels) the detection can still work reliably for objects that are close to the camera.

3.4.1.2 Region of interest

Since the input stereo data are images whose pixels are mapped to real-world coordinates an optimization can be performed by only processing the parts of the image that are of

particular interest for the detection algorithm. This approach is often called “region of interest” (ROI) in image processing.

3.4.1.2.1 Braking corridor crop

The simplest ROI optimization may be the braking corridor crop. In this optimization a region of interest for the source stereo images would be computed from the actual bounding coordinates of the braking corridor. Optical systems with wide angle lens would benefit the most from this optimization since ROI may be significantly smaller than the source stereo images.

3.4.1.2.2 Multi-pass

The idea behind multi-pass stereo matching optimization is similar to the sharp vs. peripheral vision nature of the human eye. In the first pass stereo matching is performed for a full field of view but a low resolution is be used to obtain a low resolution 3D data. The 3D data in this stage is used to detect the biggest objects and estimate their coordinates. These coordinates are reprojected back to the 2D source image and a second stereo matching pass is performed on a regions of interest in high resolution. The stereo matching over a given ROI is faster than when performed on the whole image.

3.4.2 Filtering

Point cloud filtering deals with reducing the total number of points in the cloud. Given the quadratic time complexity of the proposed detection algorithm this has a big impact on real time performance.

3.4.2.1 Point cloud cropping

The simplest way of point cloud filtering is point cloud “cropping”. This reduces the absolute number of points while still preserving a potential region of interest for the detection algorithm.

3.4.2.2 Point cloud normalization with voxel grid filter

The voxel grid filtering works by creating a cubic Cartesian 3D grid and merging nearby point in the cloud into a single point. If multiple points of the original point cloud fall into the same voxel cube they are merged into one point. The coordinates of the original points are floored to the nearest lower voxel size interval. Based on the voxel size this kind of filtering can significantly reduce the number of points in the cloud while still preserving the shape of the main objects in the point cloud.

It was found that while considerably reducing the amount of points in the cloud and thus contributing significantly to real-time performance, the object detection algorithm still performed reasonably well. However, the evaluation of distance estimation was not performed on a point cloud that was filtered by voxel grid.

The disadvantage of voxel grid filtering is that depending on the voxel size used the object detection has a limited range - the “native” pixel density in given distance from camera

based on center projection model may be lower than the actual voxel size. Informal tests based on the stereo dataset used in this thesis showed that with a voxel size of 10 cm the detection range varied from 18 to 72 meters depending on the image resolution used.

The maximum detection range in meters for point cloud filtered with voxel grid can be computed like this:

maxDetectionDistanceMeters = voxelSize * cameraF,

where

voxelSize is the size of a voxel in meters and cameraF is the focal length of the camera in pixels.

3.4.3 Detection

In the detection stage the optimization would mean using a detection algorithm that has a lower time complexity than the implemented $O(n^2)$. Tree data structures were created to lower the time complexity of the search problem in multi-dimensional spaces. An octree data structure could be used in the context of this thesis to lower the time complexity of the detection algorithm to O(n*log (n)). After such optimization the inner loop of the detection algorithm would be replaced with octree search that would be limited to the same bounding box as the inner loop.

Another type of optimization may be in using a Z-histogram that would carry the total number of points in the area in front of the camera in given distance intervals and perform the detection algorithm in a single loop. The complexity of such histogram search would be O(n). However, such histogram based detection algorithm may be less flexible in terms of detection volume size, expected object size and position and may also be susceptible to noise. Also, the road plane would need to be filtered out from the point cloud in order not to be falsely detected as object.

4 Evaluation

The detection algorithm as described above was implemented in C++ with the help of OpenCV 2.4.8 and Point Cloud Library 1.7.2. The operating system used was Ubuntu Linux 14.04 running in Oracle VM VirtualBox 4.3.12 under Microsoft Windows 7 which was run on an Intel Core i5-4570 machine with 12 GB RAM.

The performance in distance measurement of stereo vision was evaluated against LIDAR ground truth data. Three test sequences were selected for the evaluation of stereo vision based detection algorithm - one sequence with good lighting conditions, one with difficult lighting conditions and one with variable lighting conditions.

A custom point cloud crop was applied for the detection that simply filtered points that were too far on the sides or too high considering the traffic situation. The camera was placed in the centerline of the vehicle and it constituted the coordinate system center. The boundaries applied were following: 1 m to the right, 2 m to the left, 1 m to the height and 150 m into distance. The boundaries were not centered to 0 on the x axis because the leading vehicles were often offset to the left side.

The three sequences contained 1200 stereo images altogether. Since the measurements were performed for three different resolutions for each sequence, the total number of measurements made was 3600.

4.1 Input data used

The KITTI dataset was used for stereo vision evaluation. The dataset contains sequences recorded at various settings (road, city, residential etc.) in various lighting conditions. Rectified image pairs together with LIDAR data were used from the dataset. All data was logged at 10 Hz.[30]

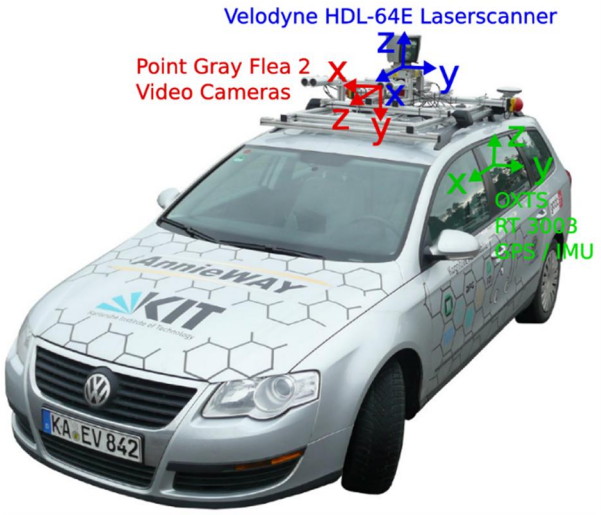

Figure 25: The KITTI capture vehicle (Volkswagen Passat B6) with the equipment on the roof. Coordinate system orientations are shown for camera and LIDAR scanner.[30]

The stereo images used were grayscale (from camera 0 and camera 1) with a native resolution of about 0.47 megapixels. Following sequences from the dataset were used for the evaluation:

- Sequence 1
  - sequence name: 2011_09_26_drive_0015_sync
  - number of frames: 297
  - lighting conditions: good
  - frames used: all
- Sequence 2
  - sequence name: 2011_09_30_drive_0016_sync
  - number of frames: 279
  - lighting conditions: difficult
  - frames used: all
- Sequence 3
  - sequence name: 2011_10_03_drive_0047_sync
  - number of frames: 624
  - lighting conditions: variable
  - frames used: 0-165, 210-570, 740-836

Figure 26: Example image from sequence 1 (frame 51).

Figure 27: Example image from sequence 2 (frame 101).

Figure 28: Example image from sequence 3 (frame 343).

Three input image resolutions were used for the evaluation according to an internal variable called pixelDivConst, which specifies the division factor for the image size (width and height) – 1 means full resolution, 2 divides image sides by 2 resulting in a resolution of $1/4^{th}$ and 4 divides by 4 resulting in a resolution of $1/16^{th}$. If the width of height of the full resolution image was not divisible by 4 the image canvas was resized to a nearest lower multiple of 4 while keeping the top left corner in its place – with this procedure the image center did not move which is important for reprojecting depth into 3D space. Here are sample pixelDivConst values with corresponding image sizes:

- pixelDivConst = 1 – 1224x368 pixels (full resolution)
- pixelDivConst = 2 – 612x184 pixels (medium resolution)
- pixelDivConst = 3 – 306x92 pixels (low resolution)

4.1.1 Stereo vision

The StereoBM method from the OpenCV library was used for stereo matching. The method has several parameters that can be set. Following parameters were used:

- common for all resolutions:
  - speckleWindowSize = 100
  - speckleRange = 100
  - preFilterSize = 5
  - preFilterCap = 63
  - textureThreshold = 0
  - minDisparity = 0
- resolution specific:
  - pixelDivConst = 1
    - SADWindowSize = 19
    - numberOfDisparities = 128
    - uniquenessRatio = 5
  - pixelDivConst = 2
    - SADWindowSize = 9
    - numberOfDisparities = 64
    - uniquenessRatio = 10
  - pixelDivConst = 4
    - SADWindowSize = 5
    - numberOfDisparities = 32
    - uniquenessRatio = 15

Following parameters for the object detection algorithm in question were used in the evaluation:

- windowSizeX = 1 m
- windowSizeY = 1 m

- minNeighbourCoef = 0.2

The stereo disparity reprojection to 3D was performed within two loops. The first loop iterated over image rows from the image bottom to the top and the second (nested) loop iterated over image columns from the left to the right.

4.1.2 LIDAR

Ground truth was provided by the LIDAR data from the dataset. The LIDAR used was the Velodyne HDL-64E which features 64 laser beams. The parameters of the LIDAR are following: [31]

- 120m range
- 360° Horizontal FOV
- 26.8° Vertical FOV
- < 2cm accuracy

Figure 29: The Velodyne HDL-64E LIDAR unit.[31]

The coordinate system of the LIDAR sensor has different orientation than the coordinate system used by the stereo camera. The sensor is also placed in a given distance from the camera. Because of these facts the LIDAR point cloud had to be transformed to the coordinate system of the stereo camera. A 3x3 rotation matrix and a 3x1 translation vector supplied with the dataset was used to construct the 4x4 transformation matrix. The transformation itself was performed by a function of the Point Cloud Library.

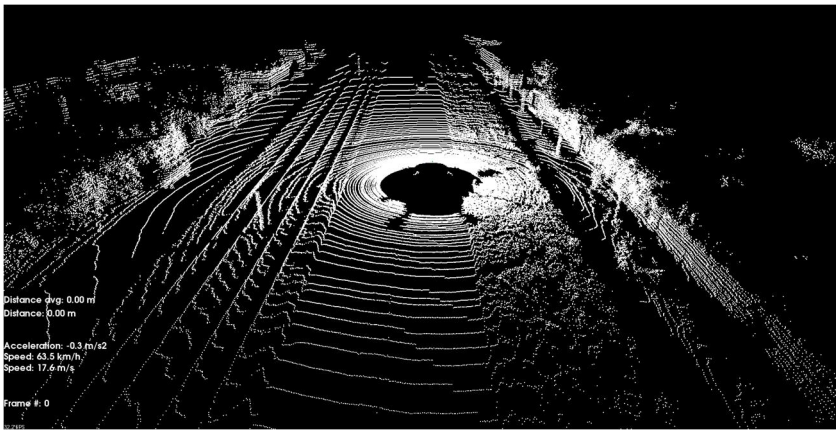

Figure 30: Example point cloud generated from the LIDAR data (sequence 1, frame 0).

4.1.2.1 Dealing with noise

Initially the intention was to detect the nearest object in the point cloud originating from the LIDAR sensor in a single loop by looking for a single nearest point in a given area/direction. However, it was found that there was a limited amount of noise present – sparse random points appeared on coordinates where no real object was present. The noise had mostly form of single points and did not allow for a single-point based distance estimation.

To overcome the problem with noise in LIDAR data few methods were considered. The first one was to measure via single point and try to discard invalid distance measurements afterwards. However, this method was found to be difficult to implement especially should the noise be present in many subsequent LIDAR frames. The second method, which was implemented, employs the same object and distance detection algorithm that is in question for the stereo vision. The parameters of the algorithm were altered to allow for effective noise suppression in the LIDAR data while possibly obtaining the best possible ground truth value. In the end, the object and distance was detected in the LIDAR point cloud using the algorithm with following parameters:

- windowSizeX = 0.5 m
- windowSizeY = 0.5 m
- minNeighbourCoef = 0.1

The detection window was made smaller with smaller point density for the LIDAR in comparison to stereo to allow returning the best possible ground truth value while filtering the noise out.

It was needed to count the point density in a given distance from the LIDAR sensor since the detection algorithm relies on it. The horizontal angular resolution of the LIDAR sensor is 0.08 degrees and the vertical angular resolution is 0.4187 degrees which gives 664.1262 laser beams on 1 m of width and 126.8778 laser beams on 1 m of height at the distance of

1 m from the sensor respectively. By multiplying these two values a total number of beams of 84262.9175 is obtained for square area of 1 x 1 m at the distance of 1 m.

The equation for the expected point density in a given distance can be obtained in following manner:

pointsOn1m2At1m = 84262.9175

pointsInWindowAt1m = pointsOn1m2At1m * windowSizeX* windowSizeY

minNeighbourCountInDistance = (minNeighbourCoef * pointsInWindowAt1m) / (zv* zv),

where

windowSizeX is the horizontal detection window size in m,

windowSizeY is the vertical detection window size in m,

minNeighbourCoef is the point density coefficient,

zv is the distance from the LIDAR sensor and

minNeighbourCountInDistance is the total absolute number of points on the area in distance zv.

4.2 Measurements

Three traffic sequences from the dataset were used for measurement. Example object detections in the first frame of the first sequence is shown on the figures below. The points what were detected in the detection window are marked with green color. All three stereo image resolutions are shown. It can be seen that the number of points drops significantly from high to low resolution.

Figure 31: Crop of frame 0 in sequence 1 (high resolution) showing the lead vehicle.

Figure 32: Object detected in stereo frame 0 of sequence 1 (high resolution) with the number of points used for detection colored in green.

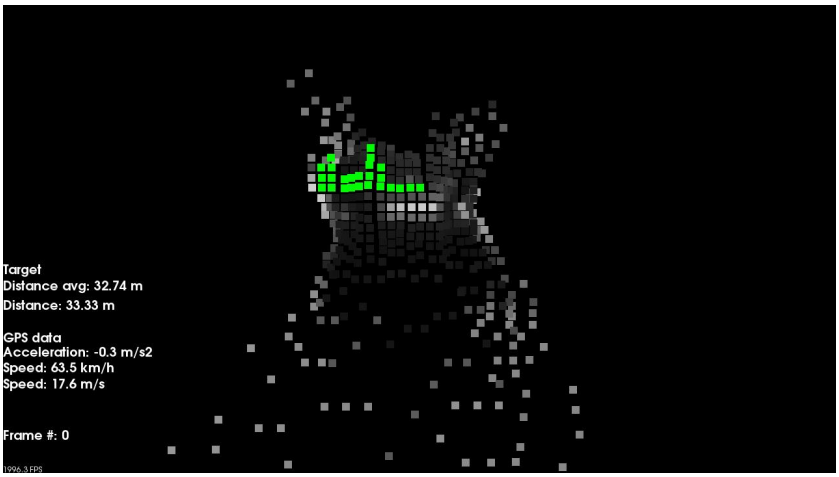

Figure 33: Object detected in stereo frame 0 of sequence 1 (medium resolution) with the number of points used for detection colored in green.

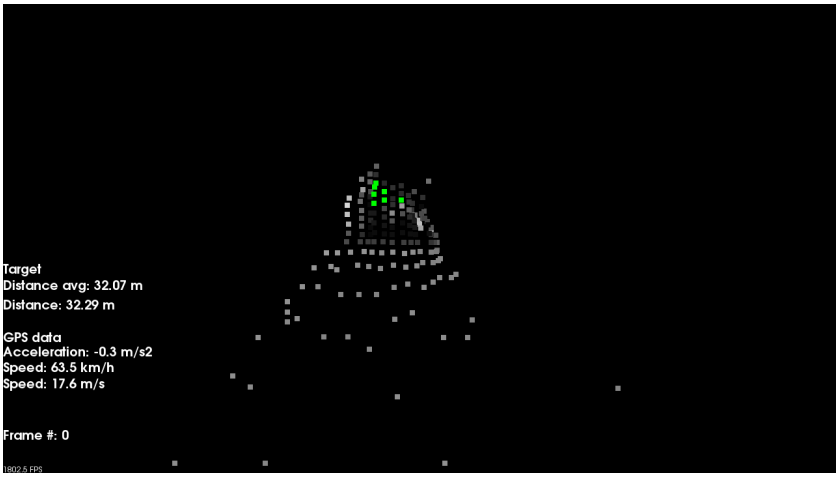

Figure 34: Object detected in stereo frame 0 of sequence 1 (low resolution) with the number of points used for detection colored in green.

Figure 35: Object detected in LIDAR frame 0 of sequence 1 with the number of points used for detection colored in green.

4.2.1 Sequence 1

This sequence has good lighting conditions and the scene is mostly lit with direct sunlight without very dark shadows. The detection results are consistent also across all resolutions.

Figure 36: Sequence 1 - first image frame.

Figure 37: Sequence 1 - last image frame.

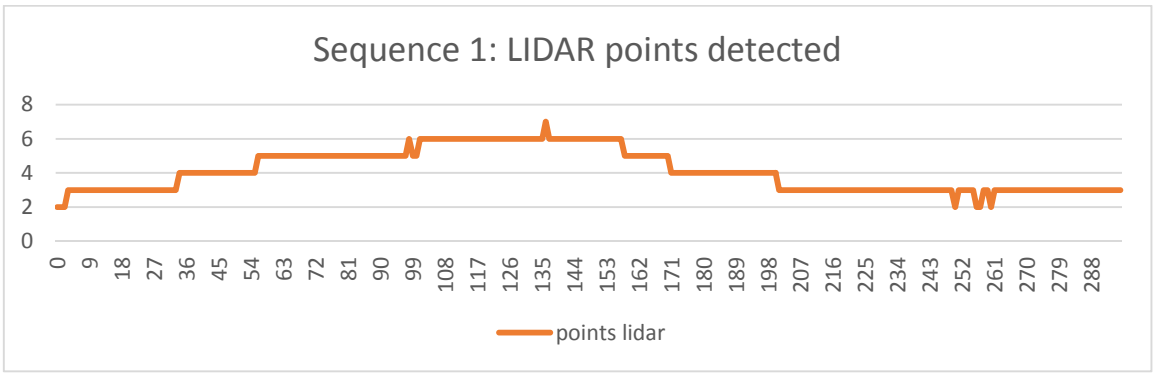

Figure 38: Sequence 1 - number of LIDAR points used in detection (y) in particular frame (x).

4.2.1.1 High resolution

Table 1: Measurement results for sequence 1 (high resolution)

Name	Value	Units
Image resolution	1224x375	pixels
Number of frames	297	pcs
Object detection rate	100	%
Average relative deviation	0.8865	%
Average absolute deviation	0.2374	m
Variance	0.0962	m^2
Maximum relative deviation	4.3718	%
Maximum absolute deviation	1.4190	m

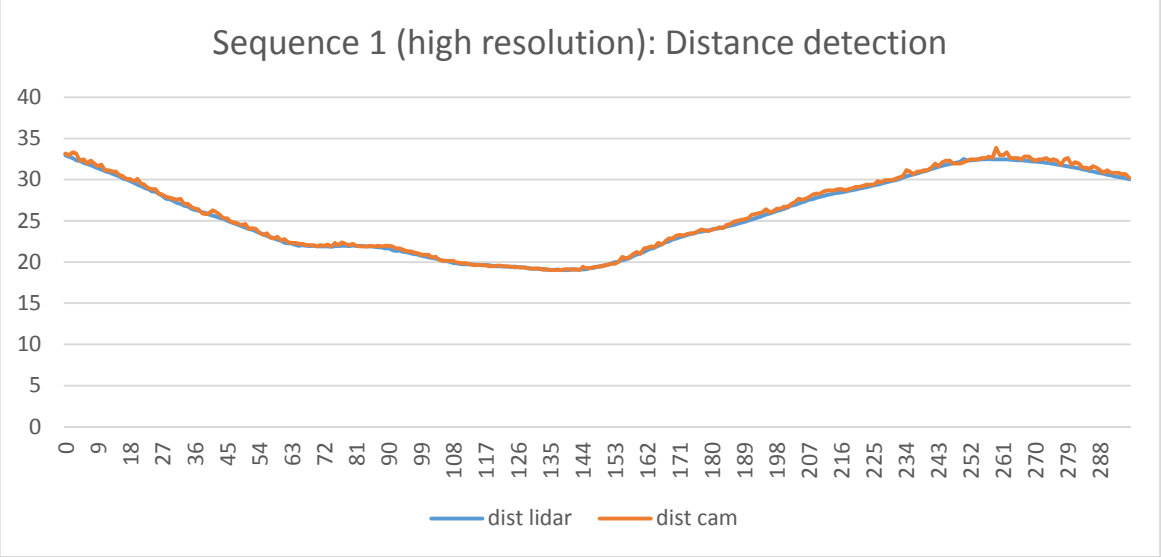

Figure 39: Sequence 1 (high resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

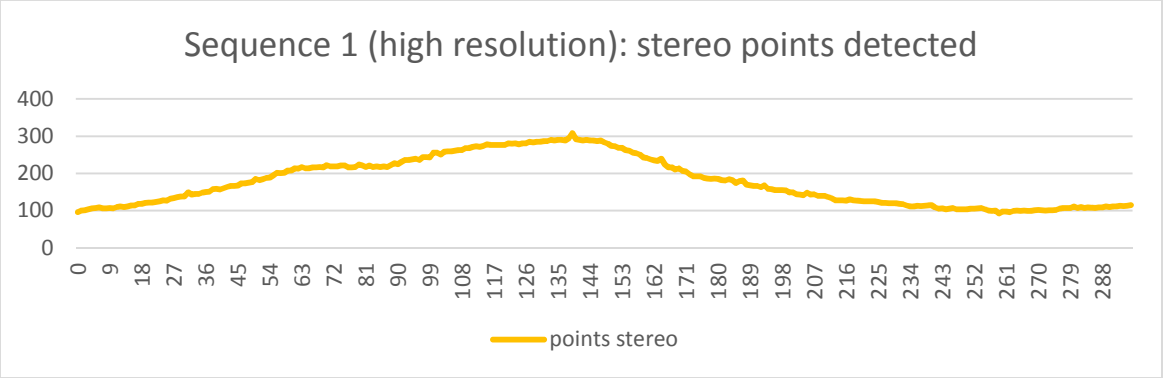

Figure 40: Sequence 1 (high resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.1.2 Medium resolution

Table 2: Measurement results for sequence 1 (medium resolution)

Name	Value	Units
Image resolution	621x188	pixels
Number of frames	297	pcs
Object detection rate	100	%
Average relative deviation	0.9062	%
Average absolute deviation	0.2455	m
Variance	0.1145	m^2
Maximum relative deviation	3.8046	%
Maximum absolute deviation	1.2349	m

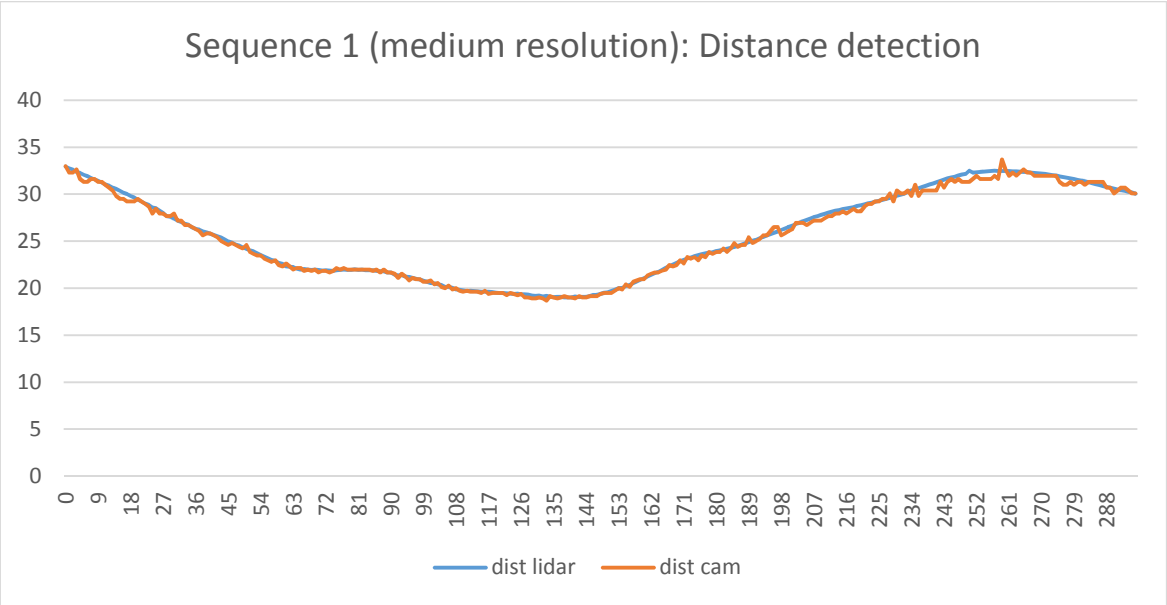

Figure 41: Sequence 1 (medium resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

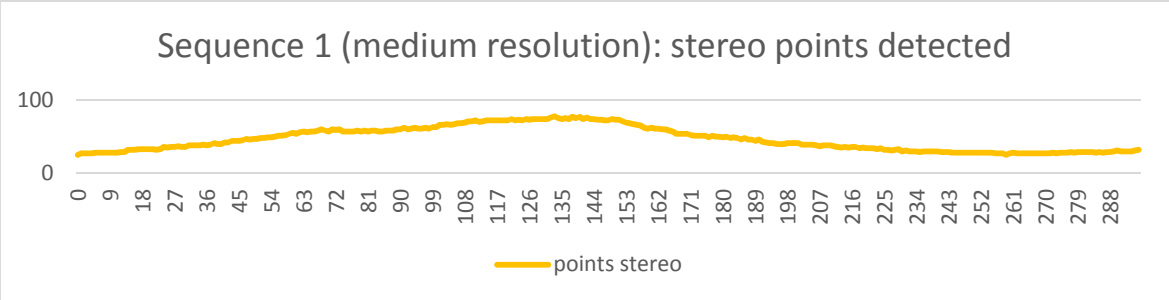

Figure 42: Sequence 1 (medium resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.1.3 Low resolution

Table 3: Measurement results for sequence 1 (low resolution)

Name	Value	Units
Image resolution	310x94	pixels
Number of frames	297	pcs
Object detection rate	100	%
Average relative deviation	3.5357	%
Average absolute deviation	0.9313	m
Variance	1.2341	m^2
Maximum relative deviation	14.9776	%
Maximum absolute deviation	4.0151	m

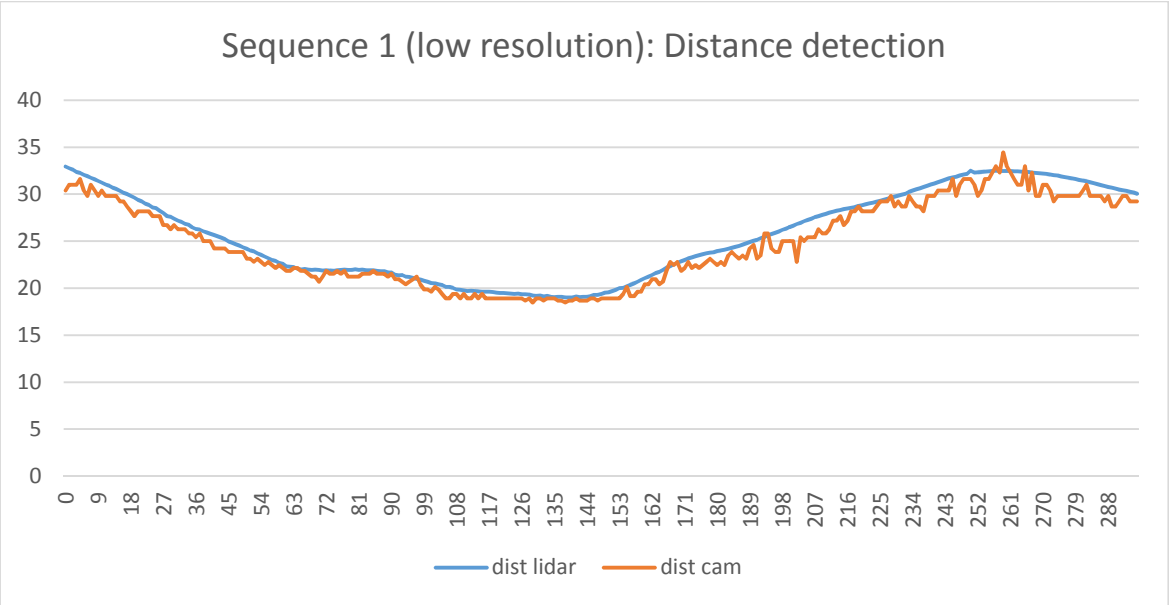

Figure 43: Sequence 1 (low resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

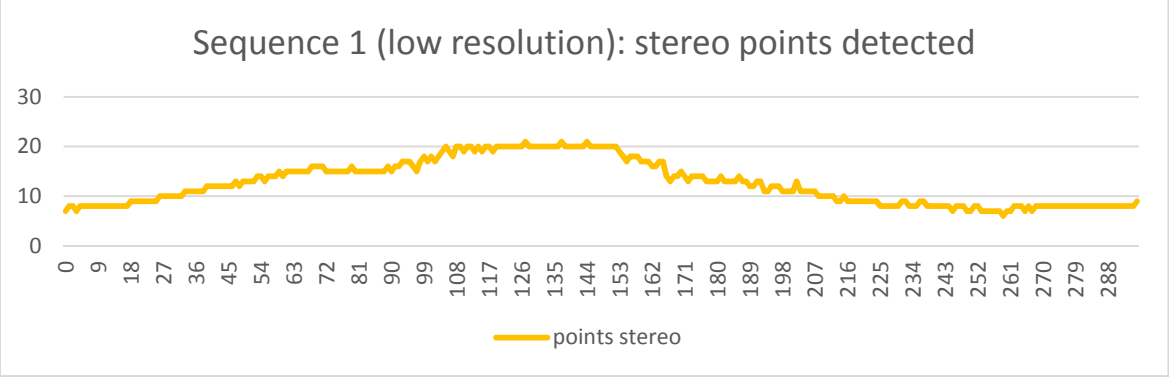

Figure 44: Sequence 1 (low resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.2 Sequence 2

This sequence presents some challenging light situations as can be seen from the last frame. Despite the lead vehicle having white color that reflects a lot of sunlight the depth of the vehicle is estimated correctly in the most frames. However, a strong reflection from the sun right after the vehicle leaves a dark shadow in the frame number 203 in high resolution image caused the StereoBM method to give erroneous depth estimation on a vehicle edge that could not be properly filtered in the detection algorithm since the point density coefficient (minNeighbourCoef) was set to a low value (0.2). A solution to this problem could either be using a better stereo matching algorithm with global disparity smoothing (like the Semi-Global Matching discussed above) or setting the minNeighbourCoef density coefficient to a higher value (0.5). Another erroneous depth estimation was present around frame number 224 in the medium resolution image for an unknown reason.

Figure 45: Sequence 2 - first image frame.

Figure 46: Sequence 2 - last image frame.

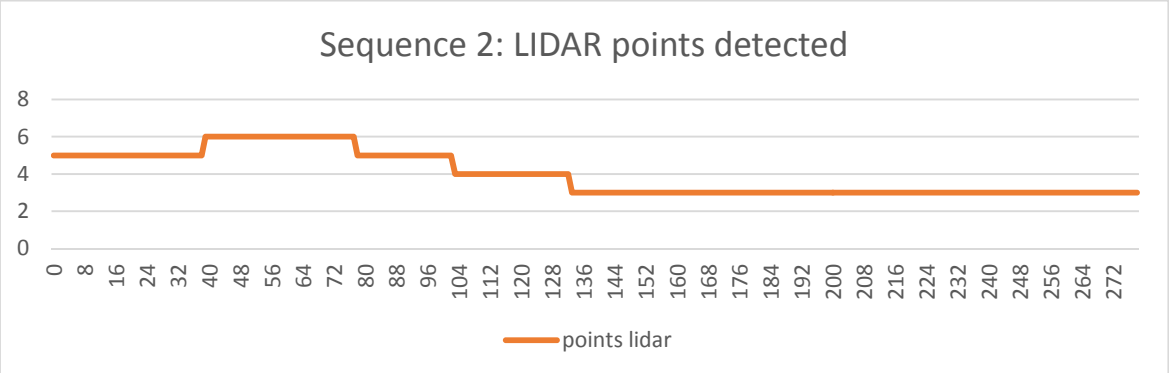

Figure 47: Sequence 2 - number of LIDAR points used in detection (y) in particular frame (x).

4.2.2.1 High resolution

Table 4: Measurement results for sequence 2 (high resolution)

Name	Value	Units
Image resolution	1224x368	pixels
Number of frames	279	pcs
Object detection rate	100	%
Average relative deviation	0.8589	%
Average absolute deviation	0.2260	m
Variance	0.1371	m^2
Maximum relative deviation	9.8400	%
Maximum absolute deviation	2.9601	m

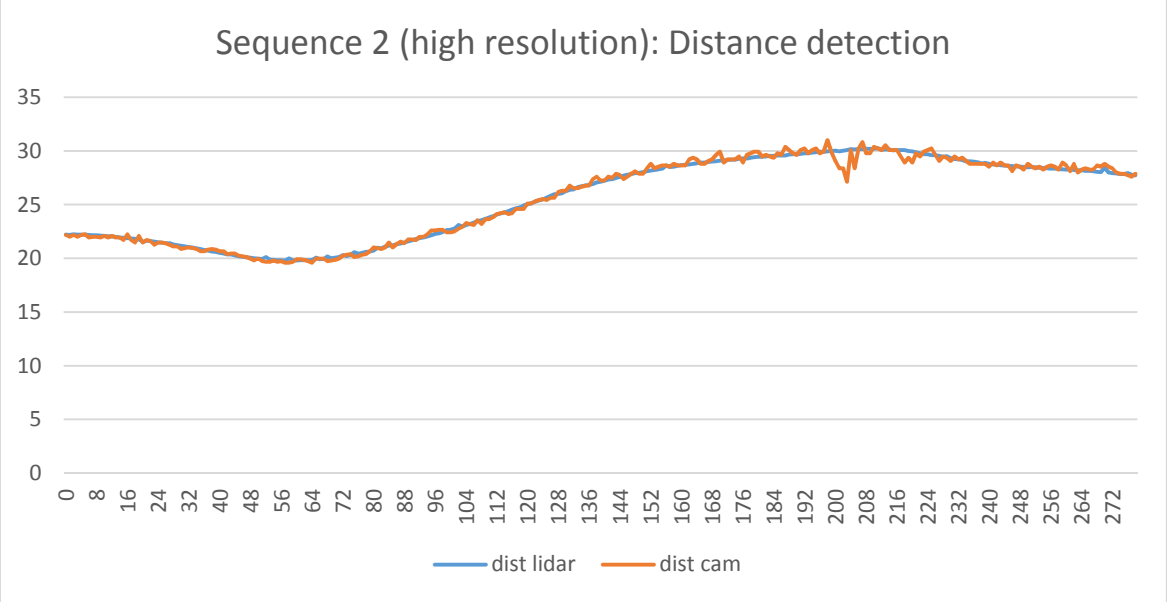

Figure 48: Sequence 2 (high resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

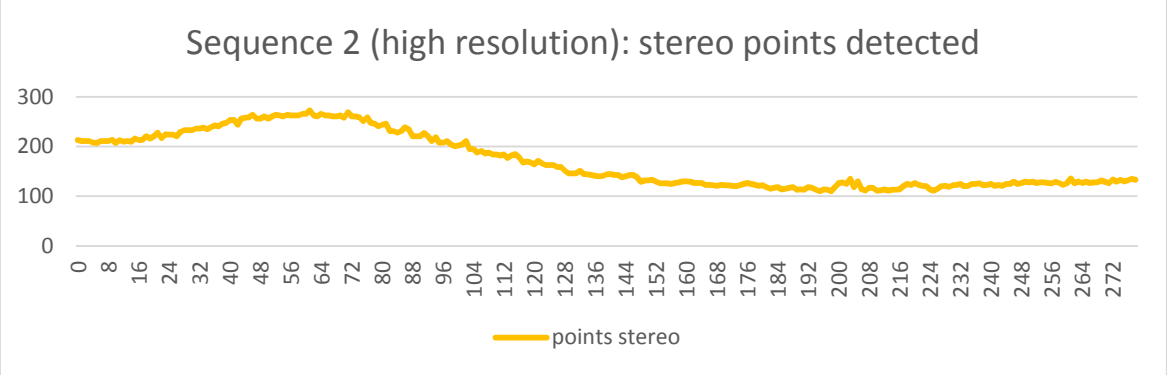

Figure 49: Sequence 2 (high resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.2.2 Medium resolution

Table 5: Measurement results for sequence 2 (medium resolution)

Name	Value	Units
Image resolution	612 x 184	pixels
Number of frames	279	pcs
Object detection rate	100	%
Average relative deviation	1.6478	%
Average absolute deviation	0.4251	m
Variance	0.4600	m^2
Maximum relative deviation	21.3049	%
Maximum absolute deviation	6.3260	m

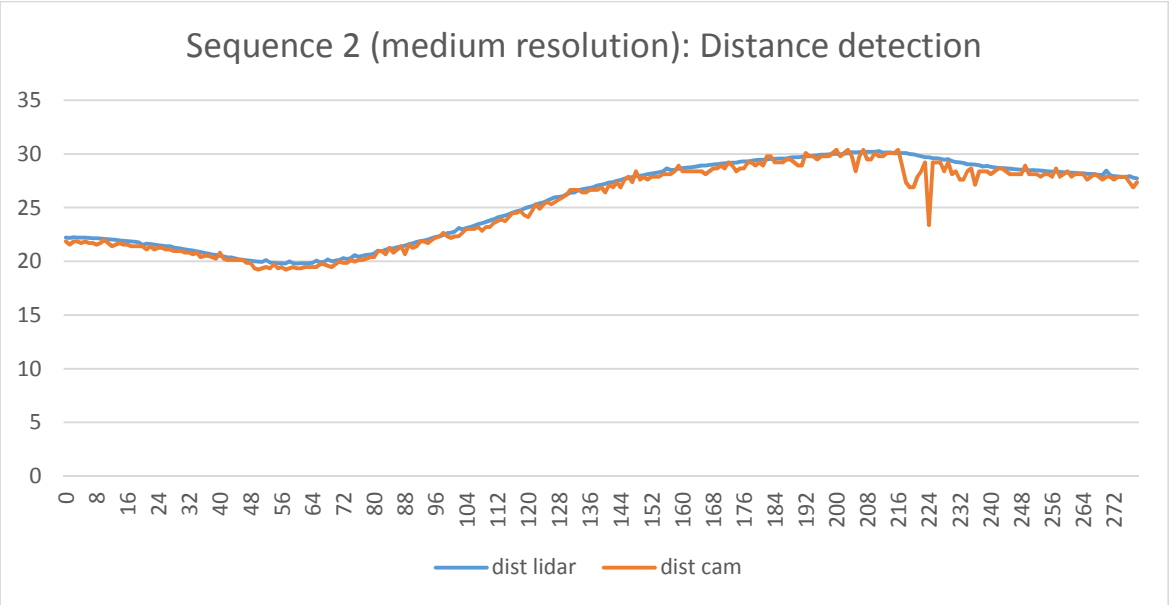

Figure 50: Sequence 2 (medium resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

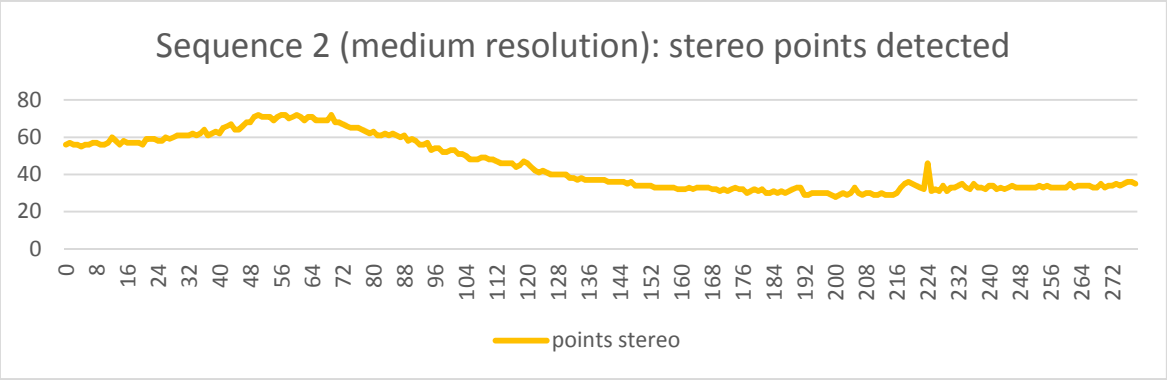

Figure 51: Sequence 2 (medium resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.2.3 Low resolution

Table 6: Measurement results for sequence 2 (low resolution)

Name	Value	Units
Image resolution	306 x 92	pixels
Number of frames	279	pcs
Object detection rate	100	%
Average relative deviation	3.2824	%
Average absolute deviation	0.8380	m
Variance	0.9713	m^2
Maximum relative deviation	9.3452	%
Maximum absolute deviation	2.6095	m

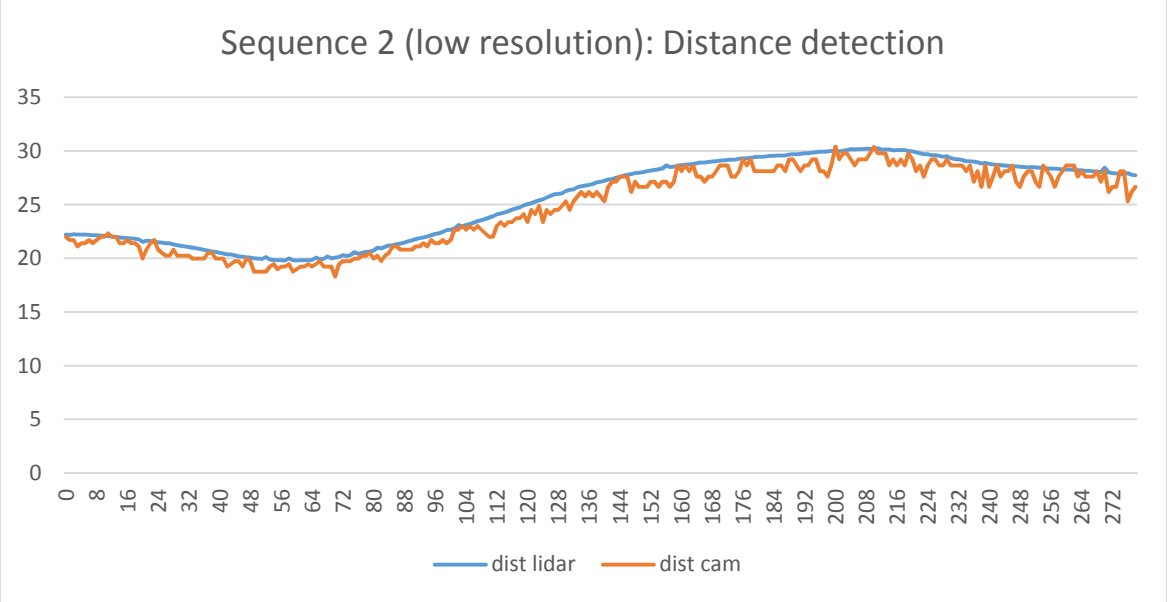

Figure 52: Sequence 2 (low resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

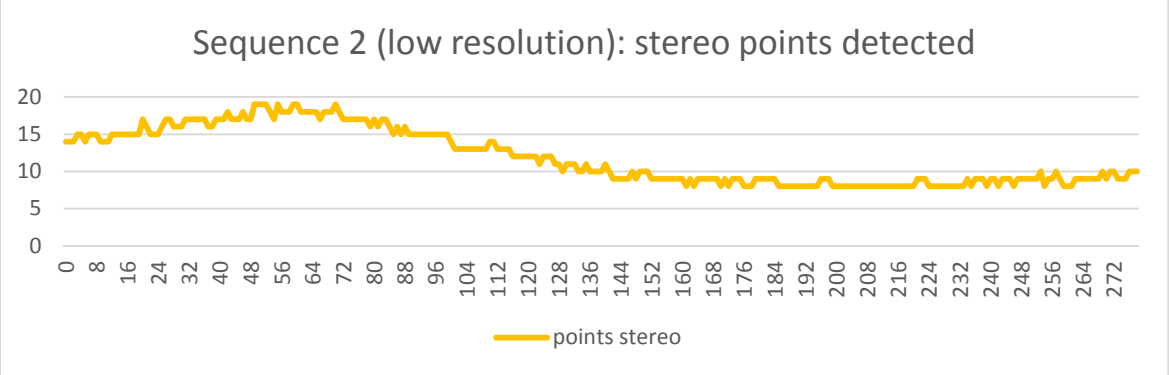

Figure 53: Sequence 2 (low resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.3 Sequence 3

This scene has somewhat challenging lighting situation with varying sunlight and shadows. However, despite this fact both the stereo matching and object detection algorithm results are consistent. The rapid distance changes between frames 165-210 and 570-740 are caused by the fact that the frames between were skipped in the detection. These frames were skipped because the lead vehicle was fully or partly missing from the considered constant driving corridor.

Note: A nearby truck was detected as the closest object in the frames #274-277 for full resolution image with the standard point cloud filtering (left boundary at 2 m), so the left boundary was changed to 1.5 m and the four distance values were measured again. This did not influence any other measurements.

Figure 54: Sequence 3 - first image frame.

Figure 55: Sequence 3 - last image frame.

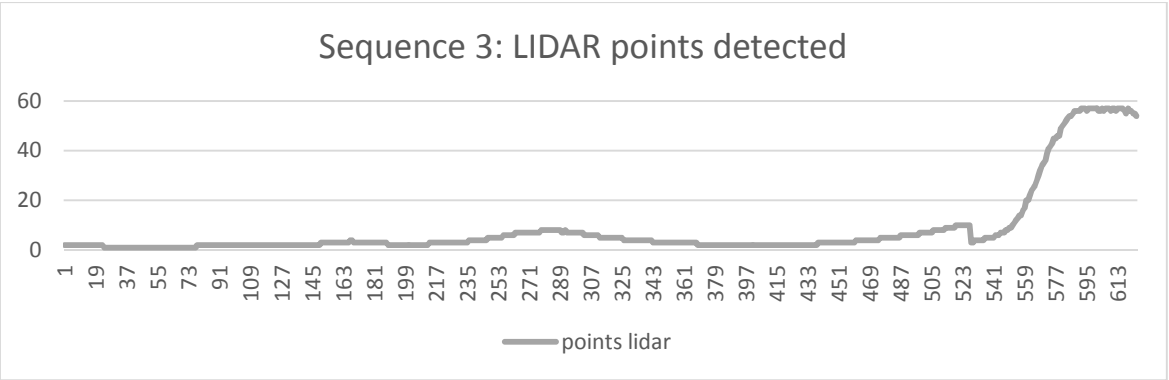

Figure 56: Sequence 3 - number of LIDAR points used in detection (y) in particular frame (x).

4.2.3.1 High resolution

Table 7: Measurement results for sequence 3 (high resolution)

Name	Value	Units
Image resolution	1240x376	pixels
Number of frames	624	pcs
Object detection rate	100	%
Average relative deviation	1.1545	%
Average absolute deviation	0.2682	m
Variance	0.1196	m^2
Maximum relative deviation	4.3745	%
Maximum absolute deviation	1.9215	m

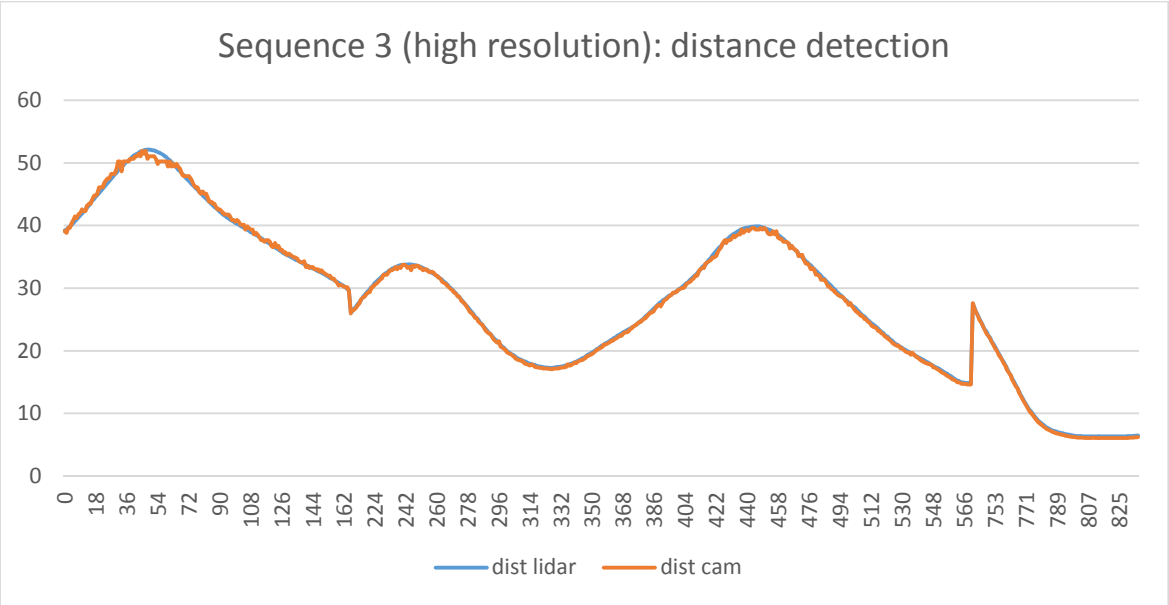

Figure 57: Sequence 3 (high resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

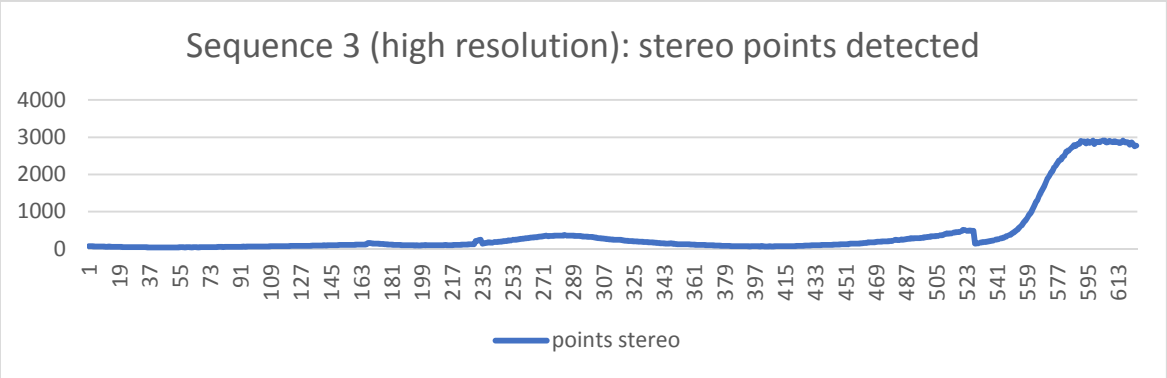

Figure 58: Sequence 3 (high resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.3.2 Medium resolution

Table 8: Measurement results for sequence 3 (medium resolution)

Name	Value	Units
Image resolution	620 x 188	pixels
Number of frames	624	pcs
Object detection rate	100	%
Average relative deviation	1.9570	%
Average absolute deviation	0.5168	m
Variance	0.4354	m^2
Maximum relative deviation	5.3296	%
Maximum absolute deviation	2.2805	m

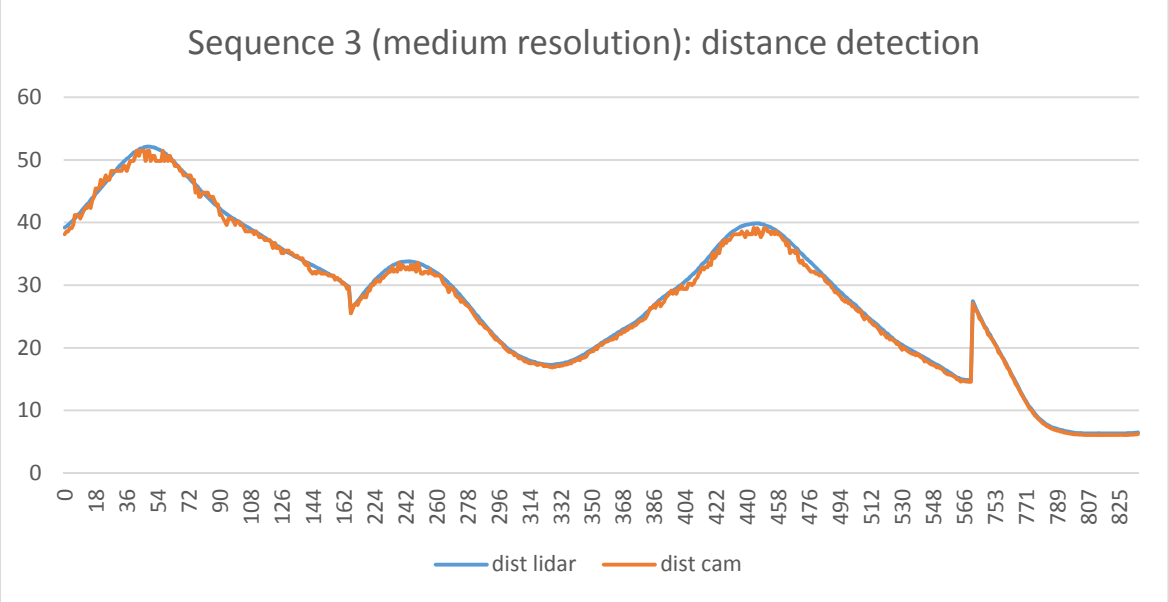

Figure 59: Sequence 3 (medium resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

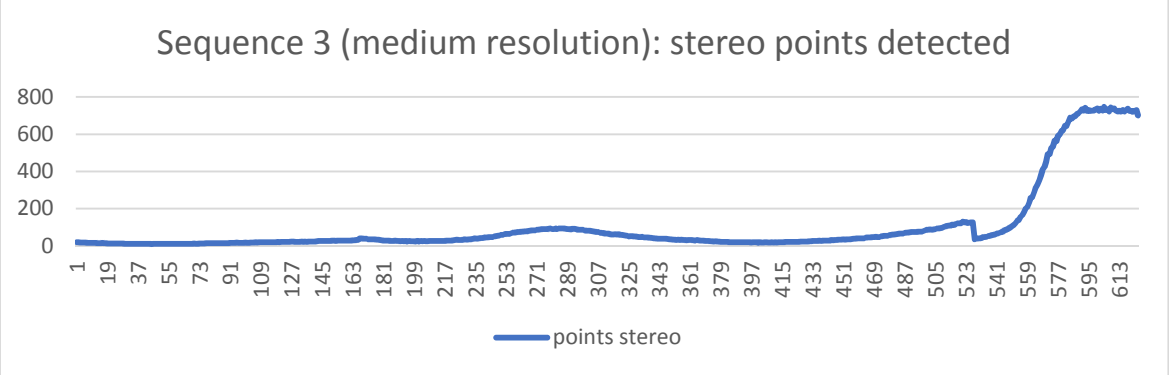

Figure 60: Sequence 3 (medium resolution) - number of stereo points used in detection (y) in particular frame (x).

4.2.3.3 Low resolution

Table 9: Measurement results for sequence 3 (low resolution)

Name	Value	Units
Image resolution	310 x 94	pixels
Number of frames	624	pcs
Object detection rate	100	%
Average relative deviation	4.3547	%
Average absolute deviation	1.2886	m
Variance	2.8385	m^2
Maximum relative deviation	12.1098	%
Maximum absolute deviation	5.314	m

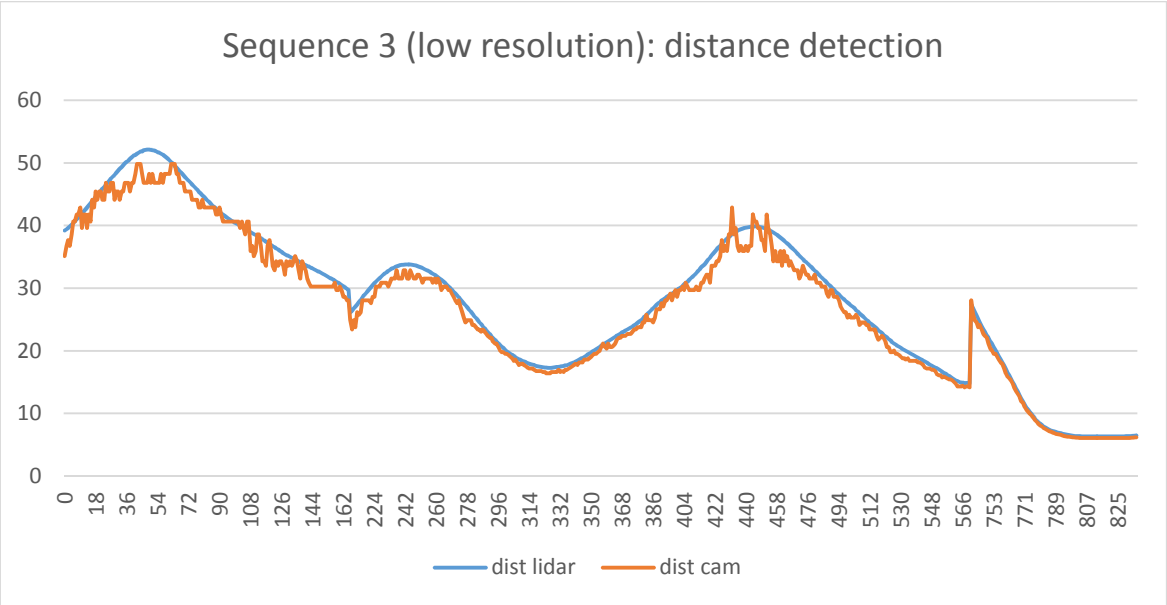

Figure 61: Sequence 3 (low resolution) – LIDAR ground truth and measured distance in meters (y) in particular frame (x).

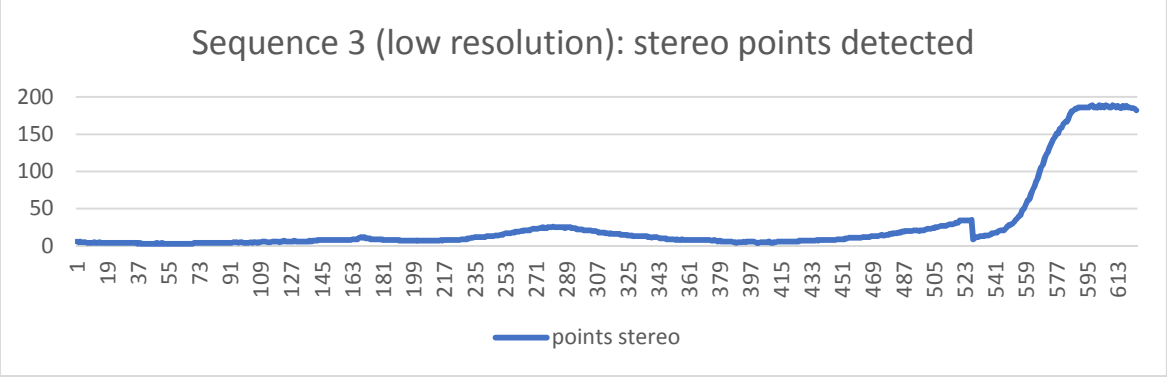

Figure 62: Sequence 3 (low resolution) - number of stereo points used in detection (y) in particular frame (x).

4.3 Results

The measurements above shown that the detection rate was very high and reached 100 % for all the sequences tested. This high detection score goes most likely back to the fact that the requirement for the sequences was that the leading vehicle will be fully visible in the camera's filed of view. Another factors that contributed to the high detection rate is the size of the target (lead vehicle) and also the size of the detection window.

The average measurement accuracies for various resolutions are shown in tables 10-12. One interesting fact is that while the medium resolution only has ¼ of the high resolution pixels the average relative deviations only grew 1.6 times, the maximum relative deviation grew also 1.6 times and variance grew 2.9 times. Similarly comparison between low and medium resolution yields average relative deviations growth of 2.5, maximum relative deviation growth of 1.2 and variance growth of 5.1. When comparing low and high resolutions (factor of $1/16^{th}$) following numbers were computed: average relative deviations growth was 3.9, maximum relative deviation growth was 2.0 and variance growth was 15.3.

The factor numbers above show that while the low resolution had only $1/16^{th}$ the pixels of the high resolution its accuracy only dropped about 4 times instead of 16 times. The used stereo matching algorithm estimates the disparity on a sub-pixel level which may contribute to the good accuracy of low resolution measurements.

Table 10: Final averaged measurement values for high resolution images.

Name	**Value**	**Units**
Average relative deviation	0.9666	%
Average absolute deviation	0.2438	m
Variance	0.1176	m^2
Maximum relative deviation	6.1954	%
Maximum absolute deviation	2.1002	m

Table 11: Final averaged measurement values for medium resolution images.

Name	**Value**	**Units**
Average relative deviation	1.5036	%
Average absolute deviation	0.3958	m
Variance	0.3366	m^2
Maximum relative deviation	10.1463	%
Maximum absolute deviation	3.2804	m

Table 12: Final averaged measurement values for low resolution images.

Name	**Value**	**Units**
Average relative deviation	3.7242	%
Average absolute deviation	1.0193	m
Variance	1.6813	m^2
Maximum relative deviation	12.1442	%
Maximum absolute deviation	3.9795	m

5 Conclusion

Object detection success rate and distance estimation accuracy of a stereo vision system were evaluated in this thesis. It was found that if the target was fully in camera's field of view and its size and detection window size were big enough then the detection rate was very high. The accuracy of high resolution images was just under 1 % which is a very good number. Interestingly the accuracy did not drop inverse proportionally with the resolution but showed better values with a factor of about 4. These findings put an interesting perspective on stereo vision system especially in respect to the low cost of such system.

Light detection and ranging (LIDAR) is generally considered to be very accurate technique for distance estimation. However, the cost and mechanical complexity of modern laser scanners are the main reasons for a slow adoption of this technology. Since a high end LIDAR can cost as much as $50,000 the room for stereo vision growth is enormous given the production cost of a stereo camera ADAS solution may be as little as $500.

The viability of stereo vision in the field of ADAS can be seen on numerous commercial systems that are available on the market today. Mercedes-Benz and Subaru have been selling stereo vision based ADAS systems as an option in multiple models since several years ago. The stereo vision system used by Subaru was able to achieve top rank in independent ADAS tests.

6 References

[1] M. Drumheller, “Mobile Robot Localization Using Sonar,” *IEEE Trans Pattern Anal Mach Intell*, vol. 9, no. 2, pp. 325–332, Feb. 1987.

[2] P. Gáspár, Z. Szalay, and S. Aradi, “Highly Automated Vehicle Systems - Environment Sensing (Perception) Layer,” 2014. [Online]. Available: http://www.mogi.bme.hu/TAMOP/jarmurendszerek_iranyitasa_angol/math-ch03.html. [Accessed: 05-Feb-2016].

[3] Continental AG, “SRL 1 Short Range Lidar (Infrared sensor),” 2016. [Online]. Available: http://www.conti-online.com/www/industrial_sensors_de_en/themes/srl_1_en.html. [Accessed: 21-Apr-2016].

[4] Delphi Automotive LLP, “Delphi Electronically Scanning Radar,” 2016. [Online]. Available: http://delphi.com/manufacturers/auto/safety/active/electronically-scanning-radar. [Accessed: 15-Apr-2016].

[5] S. Foix, G. Alenya, and C. Torras, “Lock-in Time-of-Flight (ToF) Cameras: A Survey,” *IEEE Sens. J.*, vol. 11, no. 9, pp. 1917–1926, Sep. 2011.

[6] S. Mattoccia, “Stereo vision: algorithms and applications,” *Univ. Bologna*, vol. 22, 2012.

[7] J. Yao, S. Ramalingam, Y. Taguchi, Y. Miki, and R. Urtasun, “Estimating Drivable Collision-Free Space from Monocular Video,” in *2015 IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 420–427.

[8] A. Miranda Neto, A. Correa Victorino, I. Fantoni, and J. V. Ferreira, “Real-time estimation of drivable image area based on monocular vision,” in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 63–68.

[9] Mobileye, “Mobileye’s Autonomous Car – What the System Sees,” 2015. [Online]. Available: https://www.youtube.com/watch?v=jKfwHsHUdVc. [Accessed: 29-Apr-2016].

[10] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, “Junior: The Stanford Entry in the Urban Challenge,” *J Field Robot*, vol. 25, no. 9, pp. 569–597, Sep. 2008.

[11] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, “Stanley: The Robot That Won the DARPA Grand Challenge: Research Articles,” *J Robot Syst*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[12] C. Urmson, J. Anhalt, J. A. (Drew) Bagnell, C. R. Baker, R. E. Bittner, J. M. Dolan, D. Duggins, D. Ferguson, T. Galatali, H. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, A. Kelly, D. Kohanbash, M. Likhachev, N. Miller, K. Peterson, R. Rajkumar, P. Rybski, B. Salesky, S. Scherer, Y.-W. Seo, R. Simmons, S. Singh, J. M. Snider, A. (Tony) Stentz, W. (Red) L. Whittaker, and J. Ziglar, “Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge,” Robotics Institute, Pittsburgh, PA, CMU-RI-TR-, Apr. 2007.

[13] A. Broggi, C. Caraffi, P. P. Porta, and P. Zani, “The Single Frame Stereo Vision System for Reliable Obstacle Detection Used during the 2005 DARPA Grand

Challenge on TerraMax," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 745–752.
[14] T. TerraMax, "Team TerraMax: DARPA Grand Challenge 2005." 2005.
[15] VisLab, "DEEVA's 360° stereo-based 3D perception system," 29-Apr-2016. [Online]. Available: https://www.youtube.com/watch?v=4CTZRJ-3lAU.
[16] C. Rabe, "6D-Vision," 29-Apr-2016. [Online]. Available: http://www.6d-vision.com/.
[17] Subaru of America, Inc., "EyeSight Driver Assist Technology," 29-Apr-2016. [Online]. Available: http://www.subaru.com/engineering/eyesight.html.
[18] Hitachi Brand Channel, "The New Generation Stereo Camera - Hitachi," 29-Apr-2016. [Online]. Available: https://www.youtube.com/watch?v=t7uVShyHBhk.
[19] Insurance Institute for Highway Safety, "IIHS issues first crash avoidance ratings - IIHS News," 29-Apr-2016. [Online]. Available: https://www.youtube.com/watch?v=omHES8mqtW4.
[20] W. Hulshof, I. Knight, A. Edwards, M. Avery, and C. Grover, "Autonomous emergency braking test results," in *Proceedings of the 23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2013.
[21] S. Lazebnik, "COMP 776: Computer Vision - Stereo (Lecture 13)." The University of North Carolina at Chapel Hill, 2009.
[22] N. Navab and C. Unger, "Stereo Matching." Technical University Munich, 2009.
[23] C. McCormick, "Stereo Vision Tutorial - Part I," 2014. [Online]. Available: http://mccormickml.com/2014/01/10/stereo-vision-tutorial-part-i/. [Accessed: 29-Apr-2016].
[24] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 2, pp. 807–814 vol. 2.
[25] German Aerospace Center - Institute of Robotics and Mechatronics, Perception and Cognition, "Stereo Vision," 2016. [Online]. Available: http://www.dlr.de/rmc/rm/en/desktopdefault.aspx/tabid-9389/16104_read-39811/. [Accessed: 29-Apr-2016].
[26] D. Gallup, J. M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, 2008, pp. 1–8.
[27] P. Pinggera, D. Pfeiffer, U. Franke, and R. Mester, "Know your limits: Accuracy of long range stereoscopic object measurements in practice," in *Computer Vision–ECCV 2014*, Springer, 2014, pp. 96–111.
[28] D. Lau, "LEADING EDGE VIEWS: 3-D Imaging Advances Capabilities of Machine Vision: Part I," 2012. [Online]. Available: http://www.vision-systems.com/articles/print/volume-17/issue-4/departments/leading-edge-views/3-d-imaging-advances-capabilities-of-machine-vision-part-i.html. [Accessed: 29-Apr-2016].
[29] A. Geiger, "Karlsruhe Dataset: Stereo Video Sequences + rough GPS Poses," 2016. [Online]. Available: http://www.cvlibs.net/datasets/karlsruhe_sequences/. [Accessed: 29-Apr-2016].
[30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *Int. J. Robot. Res. IJRR*, 2013.
[31] Velodyne LiDAR, "HDL-64E," 2016. [Online]. Available: http://velodynelidar.com/hdl-64e.html. [Accessed: 29-Apr-2016].

7 Appendix

7.1 Raw measurement data

Raw measurement data in Microsoft Excel format can be downloaded at the following link:

https://drive.google.com/open?id=0Bw2bJvYVnbZcODl3UzRCSjBjWFE

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2014/2015

Studijní program: Aplikovaná informatika
Forma: Kombinovaná
Obor/komb.: Aplikovaná informatika (ai2-k)

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Bc. Kučera Jan	V Zátiší 1572, Náchod	I14291

TÉMA ČESKY:

Aplikace počítačového vidění pro podporu řízení vozidel

TÉMA ANGLICKY:

Computer Vision Applications in Advanced Driver Assistance Systems

VEDOUCÍ PRÁCE:

Ing. Karel Petránek - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

1. Průzkum existujících metod počítačového vidění, volba relevantních metod
2. Průzkum problémů automobilového průmyslu, které lze řešit pomocí počítačového vidění
3. Návrh řešení, porovnání metod
4. Testování a zhodnocení výsledků

SEZNAM DOPORUČENÉ LITERATURY:

Machine Learning kurz na Coursera

Podpis studenta: ..... **Datum:** .....

Podpis vedoucího práce: ..... **Datum:** .....