CHALMERS
UNIVERSITY OF TECHNOLOGY

Self-Supervised Stereo Depth Estimation

Depth estimation in multiple environments through an adaptive CNN and IR light

Master's thesis in System, Control and Mechatronics

JONATAN NORDH
MARCUS VIKÉN

Department of Mechanics and Maritime Sciences

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021:27

Self-Supervised Stereo Depth Estimation

JONATAN NORDH

MARCUS VIKÉN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Adaptive Systems Research Group
Chalmers University of Technology
Gothenburg, Sweden 2021

Self-Supervised Stereo Depth Estimation
Depth estimation in multiple environments through an adaptive CNN and IR light
JONATAN NORDH
MARCUS VIKÉN

Supervisor: Peter Forsberg, CPAC Systems AB
Examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Master's Thesis 2021:27
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Adaptive Systems Research Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: 3D visualization of a stereo image pair taken in Gothenburg with the presented camera-rig. Depth prediction for each pixel is produced with the proposed CNN called SH-Net.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Self-Supervised Stereo Depth Estimation
Depth estimation in multiple environments through an adaptive CNN and IR light
JONATAN NORDH
MARCUS VIKÉN
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

We have developed a complete depth sensor unit with a self-supervised neural network and stereo camera. The sensor is both adaptive during usage and can work in dark and low light environments with aid from IR spotlights. Disparity estimation via stereo cameras has shown great performance in combination with neural networks during recent years. The reason is because deep learning reduces the computational effort considerably compared to previous methods. However, the existing deep learning methods do not evaluate the depth measurements but rather the disparity estimation accuracy on available benchmark datasets. In difference to earlier work, this system has been evaluated with respect to depth measurement accuracy and suitable evaluation metrics have been developed. If the stereo camera is to be used as a reliable depth sensor the depth estimation quality needs to be ensured. From the thesis contributions a high-functional depth sensor unit can be developed with potential to surpass other sensors with respect to the amount of data obtained per second.

Keywords: Artificial Neural Network (ANN), Convolutional Neural Network (CNN),deep learning, self-supervised, machine learning, stereo vision, disparity estimation, night-vision, Oriented FAST and Rotated BRIEF (ORB).

Acknowledgements

This master's thesis completes our studies within the M.Sc. programme in Systems, Control and Mechatronics at Chalmers University of Technology. The thesis project was carried out during the spring of 2021 at CPAC Systems AB. We would like to send our gratitude to our supervisor Peter Forsberg who has guided us through this project with great commitment. Then we would also like to thank CPAC that provided all the necessary resources required to make this project possible. Although a pandemic has made everything more complicated this year, Peter and all others at CPAC have made the best out of the situation, for this we are very impressed and grateful.

Jonatan Nordh and Marcus Vikén, Gothenburg, May 2021

Thesis advisor: Peter Forsberg, CPAC Systems AB
Thesis examiner: Peter Forsberg, Department of Mechanics and Maritime Sciences

Abbreviations

AD .....Autonomous Driving
ADAS .....Advanced Driver Assistance Systems
BRIEF .....Binary Robust Independent Elementary Features
CNN .....Convolutional Neural Network
FAST .....Features from Accelerated Segment Test
FIR .....Far Infrared
FOV .....Field Of View
FPS .....Frames Per Second
GBG .....Gothenburg
GPU .....Graphics Processing Unit
IR .....Infrared
LiDAR .....Light Detection And Ranging
LWA-Net .....Light-Weight Adaptive Network
NIR .....Near Infrared
ORB .....Oriented FAST and Rotated BRIEF
SH-Net .....Stacked Hourglass Network
SIFT .....Scale-Invariant Feature Transform
SSIM .....Structural Similarity Index Measure
ReLU .....Rectified Linear Unit
XCNN .....Cross Convolutional Neural Network

Contents

List of Figures

List of Tables

1
Introduction

The demand for autonomous and remotely controlled vehicles is growing at a rapid pace. Autonomous systems can both increase safety and reduce cost. If a human driver can be replaced in a hazardous and inhospitable environment, that would decrease both labor cost and injury risk of the driver. Areas where autonomous systems can be applied are many such as: boat docking [1], garbage collection [2], mining [3] and self-driving cars [4].
To be able to interpret the surroundings and make suitable decisions, the system of an autonomous vehicle needs accurate and dependable sensors. Malfunctioning or failure of such systems can lead to irreversible damage which consequently makes them safety critical [5]. One of the most universal signal types that is used to interpret the surrounding is the 3D depth measure whereby, producing a 3D grid map nearby objects can be located and measured. One type of sensor that can produce 3D depth estimation is the stereo camera. Here, by utilizing disparity between objects in two images, depth can be calculated. Disparity is defined as the pixel coordinate difference between an object's position in the left and right image produced by a stereo camera. Stereo cameras can produce depth measurements with higher resolution compared to LiDARs and provide color images. In addition, the cameras are inexpensive which consequently makes them competitive as depth sensors. However, current stereo cameras have a few drawbacks due to a variety of real-world problems such as occlusions, large textureless areas, reflective surfaces and insufficient light. To make accurate estimations the stereo cameras also need initial and frequent calibration due to imperfect assembly or geometric deformation from for example thermal changes. If these drawbacks were compensated for the stereo camera could be a competitive sensor within the AD/ADAS industry where a lot of data can be obtained from images in a variety of environments. The stereo camera could also be combined with infrared (IR) light and other sensors which can boost predictions during night and low light conditions.
It has been shown that by using four stereo cameras with different baselines, the accuracy can be increased for a long range, but at the cost of high computational demand [6]. Further, using deep learning the computational effort can be decreased considerably [7]. Currently, state of art stereo methods on the KITTI stereo benchmark [8, 9] leaderboard is based on deep learning. However, most of these stereo methods are using supervised learning which require ground truth data. The cumbersome labor of collecting ground truth data has here been solved by using self-supervised learning. This is a solution that also enables network adaption to new environments based on the images collected by the stereo camera during usage. Instead of developing new competitive network architectures the focus of this thesis

work is to evaluate real-life performance of networks inspired by already existing state-of-art network architectures. The performance must be evaluated in a trustworthy manner before it can be used as a depth sensor in safety-critical applications. Evaluation on data gathered with other camera configurations in different environments will not provide a reliable evaluation. Therefore, the purpose of this thesis is to present smart ways to evaluate real-world depth estimation accuracy on self-collected data and to discuss areas where the camera sensor potentially could be implemented.

1.1 Purpose

The purpose of this master's thesis is to implement and evaluate existing state-of-the-art CNN strategies for instantaneous disparity calculation using a binocular stereo camera. With a focus on real life online performance an evaluation will be done that evaluate to what extent the stereo camera can complement or replace current depth sensors. Hence, the precision and robustness of depth estimations will be examined in different environments and conditions. The aim is to address the following objectives: i) Implementation of two or more state-of-art inspired CNN architectures, ii) Combine CNN and stereo camera as a complete depth sensor unit, iii) Evaluation of full-HD depth estimation, iiii) Evaluate night vision performance, iiiii) Improve real life robustness and adaptability. This will be achieved by implementing existing theory and research and then by examining the possibilities of refining the estimation.

1.2 Objectives

The main objective of this thesis is to evaluate the stereo camera as a depth measurement sensor. This includes implementation of state-of-the-art algorithms, collect and evaluate depth data and test possible setups for usage on moving vehicles. Furthermore, the objective is to evaluate the systems to find an effective and usable system configuration.

The thesis will aim to answer the following questions:

- How can existing theory and research be implemented to design a depth estimation sensor for real-life usage?
- How can depth accuracy be evaluated on collected data?
- What is the optimal baseline distance for the proposed depth estimation sensor?
- What is the precision of estimated dense full-HD depth maps?
- How can self-supervised learning be used to enable an adaptive behavior?
- How can IR light be implemented to increase performance in low-light conditions?

1.3 Scope

The goal of this project is not to achieve good benchmark scores for any available benchmark dataset. The project will focus its resources on evaluating the areas related to depth measurements and practical implementations on board vehicles. The thesis aims at showing the potential of stereo cameras to be used as sensors and investigate the possible benefits and drawbacks of such a sensor. Therefore, efforts will not be spent trying to reach the best publicly available benchmark scores. Neither will time allow to try all network architectures such that state-of-the-art predictions are obtained. Instead, the components of chosen neural networks will be evaluated and enhanced with respect to obtaining as good dense depth estimations as possible. Faster computational speed and more advanced algorithms will most likely be presented in the future. However, this work will provide information of how to make a stereo camera useful as a depth sensor.

1.4 Related work

In recent years significant improvements have been achieved within the area of stereo vision due to the use of deep learning. The reason for this is that deep learning reduces the computational effort considerably compared to previous methods. In 2017, Kendall et al. [7] proposed the Geometry and Context network (GC-Net) which is an end-to-end disparity regression learning architecture. The network architecture is based on a Siamese network which learns deep unary features through a number of 2D convolutions. The deep unary features are then used to compute a stereo matching cost by forming a 4D cost volume using 3D convolutions. The GC-net is a state-of-art network and since it was released in 2017 many well performing networks have been designed based on this architecture. PSMNet [10] is a more recent network that further increases accuracy by introducing a stacked hourglass 3D convolution architecture. The number of 3D convolutions can be increased considerably without affecting computational cost due to frequent down and up-sampling. The PSMNet later inspired the GA-Net [11] which is replacing the computationally costly and memory-consuming 3D convolutions by introducing two new neural network layers. Mayer et al. [12] proposed a network architecture called DispNet that applies the optical flow estimation concept to disparity estimation using convolutional neural networks.

The performance of these methods are state-of-art with impressive results on popular benchmark suites like KITTI Stereo 2012, 2015 [8, 9]. However, they are all using supervised learning which require ground truth depth data. As is well known, ground truth data is often expensive and time consuming to obtain, and thus self-supervised learning is preferred.

Inspired by DispNet, Godard et al. [13] proposed a method to perform monocular depth estimation as an image reconstruction problem. By implementing an image reconstruction loss with a left-right consistency check the network can learn to perform single image depth estimation, despite the absence of ground truth data. They show that this method even outperforms supervised methods. Stereo match-

ing is closely related to monocular depth estimation and Zhong et al. [14] introduced a self-supervised learning method for stereo matching. The network predicts dense disparity maps directly from the stereo input which enables the network to be self-improving and adaptive to new unseen imageries and different camera settings. Reconstruction loss is the most common way to remove the dependency of depth ground truth data but one drawback when reconstructing right input image from left is that the network cannot handle occluded regions. Peng et al. [15] introduced an occlusion aware self-supervised stereo method. By making use of geometry features of the disparity maps in an iterative way occluded pixels can be detected and added to an occlusion mask. The resulting occlusion mask is then used as a guidance in either training or post processing. What is common for these state-of-art depth estimation algorithms is that they often ignore limitations of GPU memory space and power consumption. Gröndahl et al. [16] introduced the XCNN network and shows that the speed, GPU memory and power consumption can be decreased considerably by applying weight pruning but at the cost of network adaptability. Gan et al. [17] propose a light-weight network for real-time adaptive stereo depth estimation which is suitable for an embedded device such as NVIDIA Jetson TX2 [18].

2
Theory

In this chapter the theory specific for the thesis will be presented. Initially a brief introduction to the stereo camera and the correlation between disparity and depth is given. Thereafter theory related to the neural networks used in the project is presented. Finally, IR light for night vision and the feature matching algorithm ORB are introduced.

2.1 Binocular disparity

Humans perceive the world in three-dimensional coordinates although the human eye can only extract information in two dimensions which is made possible using binocular disparity. An object point in space appears at distinct positions for the left and right eye. The difference is called a disparity and it is proportional to the distance to the object. Objects located nearby give large disparities while objects further away result in smaller disparities. With the use of two horizontally aligned cameras a computer can estimate depths like humans do [19].

2.1.1 The stereo camera

A stereo camera consists of two or more pinhole cameras with different field of perception. In Figure 2.1 there is a representation of the pinhole camera. The light reflected from objects pass through a small aperture and projects an upside-down image on the opposite side of the box, called the image plane.

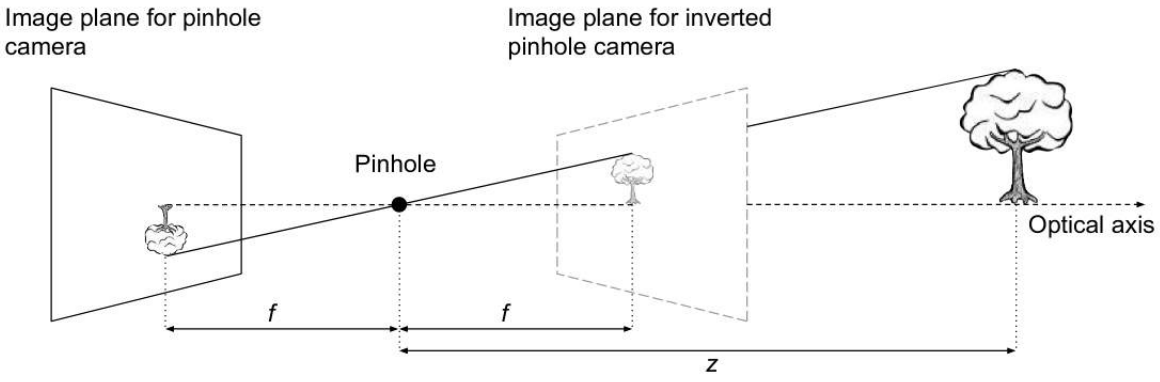


Figure 2.1: The pinhole camera.

The distance between the aperture and respective image plane is called a focal length, denoted as f. The focal lengths in the x and y directions are different, f_x and f_y, since the shape of individual pixels in a camera often are non-square. Furthermore, the possible offset between the optical axis, also called principal point, and the center of an image are expressed with the two variables c_x and c_y. Where c_x is the horizontal offset and c_y is the vertical offset to the true image middle point. From the real world-point $P = [X, Y, Z]^T$ the camera coordinates, x and y, can be calculated with the following equations:

$$x = f_x \frac{X}{Z} + c_x \tag{2.1}$$

$$y = f_y \frac{Y}{Z} + c_y \tag{2.2}$$

These relations can be expressed as a 3x3 matrix that maps between real-world coordinates and camera coordinates and is called the *intrinsic matrix*, denoted as $\boldsymbol{M}$. The transformation is written as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{2.3}$$

In many applications it is useful to transform camera coordinates to a real world coordinate system which can be achieved with linear transformation using the *extrinsic parameters*. They consist of a rotational 3x3 matrix $\boldsymbol{R}$ and a 3x1 translation vector $\boldsymbol{t}$. The mapping between world and camera coordinates is

$$\boldsymbol{P}_{camera} = \boldsymbol{R}(\boldsymbol{P}_{world} - \boldsymbol{t}) \tag{2.4}$$

Image distortion is a common effect for all cameras with a lens and there exist two common types of distortions. Firstly, radial distortion, which is an effect of having a convex lens that bend light more at the edges than in the center, resulting in a "Fish-Eye" effect. A phenomenon that can be compensated for with the following equation:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.5}$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.6}$$

The second most common lens distortion is tangential distortion as an effect of the lens not being parallel to the imaging plane. This can also be compensated for with use of these equations:

$$x_{corrected} = x + 2p_1 xy + p_2(r^2 + 2x^2) \tag{2.7}$$

$$y_{corrected} = y + p_1(r^2 + 2y^2) + 2p_2 xy \tag{2.8}$$

Here, $r = \sqrt{x^2 + y^2}$ is the pixel coordinate distance to the origin. The radial distortion coefficients k_1, k_2 and k_3 as well as the tangential distortion coefficients p_1

and p_2 are also considered *intrinsic parameters*. Even though these two distortions have the largest impact there exist other types of distortions with less impact that usually can be neglected.

2.1.2 Stereo calibration

Stereo calibration is essential for disparity calculations. A horizontal miss-alignment will contradict the assumption that matching points from the left and right camera will appear on the same horizontal line. Factors for a miss-alignment are camera and lens displacement. Either as a translation shift internal in the camera or external between cameras, such a shift is most probably quite small. However, rotational shift has a larger impact and is more difficult to notice. A small pitch-angle offset can give a large image offset in pixel-distance. It has been shown that the errors corrupt the depth estimation but there are ways to compensate for these imperfections [20]. One way to simplify the depth estimation is by calibrating the cameras. A single camera calibration makes use of the intrinsic parameters to compensate for the possible distortions. However, for a stereo pair to be identical in all aspects except the horizontal shift a stereo calibration is necessary. With the use of a known point in space visible to both cameras a relationship between the cameras in space can be calculated. The calibration is done by calculating the rotational matrix and translation vector between two cameras, calculated as:

$$\boldsymbol{R} = \boldsymbol{R_r}\boldsymbol{R_l}^T \tag{2.9}$$

$$\boldsymbol{t} = \boldsymbol{t_r} - \boldsymbol{R}\boldsymbol{t_l} \tag{2.10}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ denotes the rotation and translation to move the right camera coordinate system into the left one. To transform a point from the left to the right camera Equation 2.4 can be used with $\boldsymbol{P}_L$ instead of $\boldsymbol{P}_{camera}$.

2.1.3 Disparity and depth calculations

Once a binocular camera is stereo calibrated there is a known relation between the left and right view. A world point P should be located at the same horizontal line in both views such that they have the same y-coordinate. In opposite, the x-coordinate should have different values for the two views where the disparity is linearly related to the distance of the point P. As illustrated in Figure 2.2 the world point P results in the image points $p_L = [x_L, y_L]$ and $p_R = [x_R, y_R]$ in the left and right image, respectively. As an effect of the horizontal displacement, also known as the baseline B, the image coordinates are not identical. If perfectly stereo calibrated the vertical coordinates are the same, $y_L = y_R$, while the horizontal coordinates are not, $x_L \neq x_R$, except for points extremely far away from the camera. The disparity d is the horizontal coordinate difference of where the point P appear in respective camera, $d(p_L) = x_L - x_R$. The geometric relation of a pinhole camera gives a linear relationship between disparity and depth as

$$Z = \frac{Bf}{d(p_L)} \tag{2.11}$$

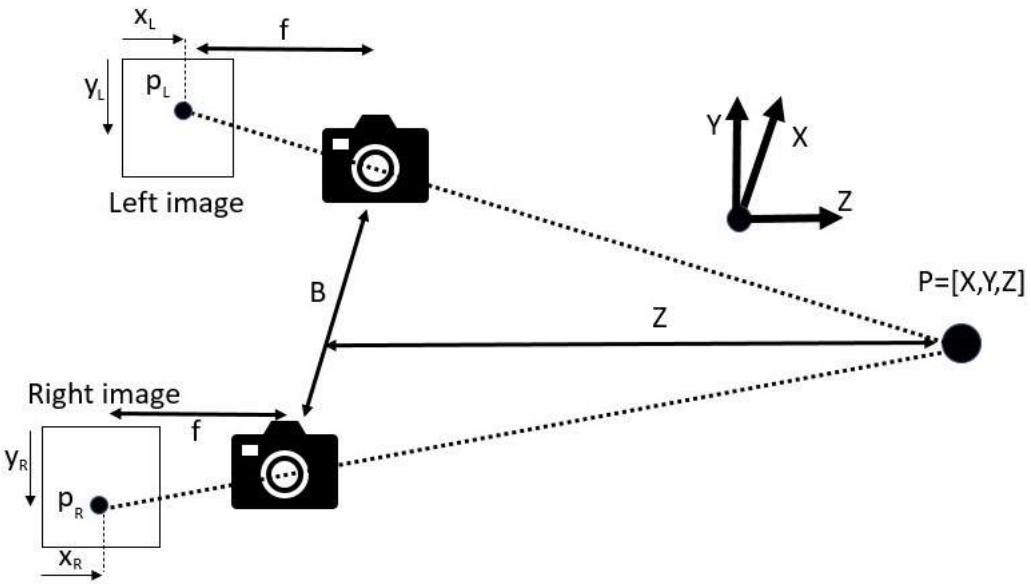


Figure 2.2: Illustration of the camera setup and disparity to depth relation in a three-dimensional space.

where the baseline B [millimeters], focal length f [pixels] and $d(p_L)$ [pixels] give the distance Z to point P in millimeters.
There exist multiple algorithms for stereo depth estimation with the common blocks:

- Matching cost computations
- Cost aggregation
- Disparity computation and optimization

The traditional matching cost algorithms are computationally heavy and/or dependent on exact stereo calibration. Since two points must be matched the size of the search area will decide the computational cost. Two of these algorithms are semi-global matching and mutual information series (SGM and SGBM) [21]. It has been proven that with the use of neural networks dense disparity-maps can be estimated faster and less dependent on perfect calibration [7].

2.2 Artificial neural networks

Artificial neural networks have shown impressive performance for the disparity estimation task. The theory and specific functions used will be presented in this section.

2.2.1 Network architecture

Different tasks need different network structures and sizes. In general, a larger/deeper network can learn more complex patterns but at the cost of being more computationally heavy. A popular network architecture is the U-net with convolutional layers down-sampling the input to a latent space and then use up-sampling

with transposed convolutional layers to the original input size [22]. The down-sampling part is called an encoder while the up-sampling part is known as a decoder. A way to not lose information while going deep into the network is by introducing residual connections [23]. These connections feed forward information from layers in the encoder to layers in the decoder adding a contribution in addition to the previous layer in the decoder. For a layer L_N^D in the decoder the contribution can be written as $L_N^D = L_{N-1}^D + wL_N^E$ where w is a weight constant and L_N^E is a layer output from the encoder. For these connections to work the layer output sizes of the decoder and added encoder layer need matching height, width and number of channels. Lastly, to extract information with respect to the difference of two input images cross-connections have proven useful. Consider two identical networks with different input data. In each layer information from the other identical network is added making it possible for the network to do comparisons between both inputs. The addition makes every layer in the left and right lane dependent and in that way forced to share trainable weights. A structure commonly known as the Siamese network architecture [24].

2.2.2 Training the network

There are two main approaches of training a neural network: supervised and self-supervised training. Supervised training use annotated data containing the ground truth of what the network tries to learn. For disparity estimation the ground truth data is the true disparity/depth map obtained from for example a LiDAR. The goal is to create a network capable of estimating disparity from a stereo pair it has never seen before. While supervised training needs a lot of manual prepossessing the self-supervised fashion is less demanding. A calibrated stereo pair of images as input is all the network need in order to learn. Instead, the complexity lies in the loss-function. Rather than minimizing a simple mean square error (MSE) between the ground truth and network output a more sophisticated loss must be defined.

2.2.3 Loss-function

All neural networks have an objective function which they try to minimize or maximize. It is the loss function that control how well the network is performing and give information about what changes that are most effective to get closer to the objective. This is done by backpropagation [25] that computes the gradient of the loss function with respect to the weights and provide a direction of change most optimal for the current state. How the loss-function is defined will therefore have significant impact on the learning and outcome of neural network training.

2.2.3.1 Supervised loss

The supervised loss can be expressed as the MSE of the predicted and ground truth disparity.

$$L_s = \frac{1}{NM} \sum_{i}^{M} \sum_{j}^{N} (d_L(i,j) - \hat{d}_L(i,j))^2 \tag{2.12}$$

for images with a shape of $[N \times M \times 3]$, predicted left disparity d_L and ground truth $\hat{d_L}$. With the objective to minimize a MSE loss-function the network will learn how to create the ground truth from the provided input. Hence, the quality of ground truth data become important as bad data will result in a poor training. The network will never perform better than the quality of the ground truth data.

2.2.3.2 Self-supervised loss

The self-supervised loss is not dependent on ground truth data to indicate performance and direction for a network. Instead, the input data can be processed in a way such that deficient performance increases the loss and superior performance result in lower loss. For disparity estimation the advantage of having a stereo image pair is utilized. The network processes the left and right input images and predict a disparity map that equals the pixel distances between the two images used to move pixels from one image to the other. For a pixel coordinate $[u, v]$ the left image has a pixel intensity value $I_R(u, v) = [R_L, G_L, B_L]$ for a three-dimensional image. The network estimates a disparity at the same position as $d_L(u, v) = \tilde{d}$ after having processed the stereo pair. Then the pixel value from the left image is copied to the position $[u + \tilde{d}, v]$ in a new image. The pixel value is compared with the same position of the right image to see if the pixel value I_R was moved correctly, called a *warping-loss* or reconstruction error.
The *warping-loss* is defined as

$$L_W = |I_R(u, v) - I_L(u + \tilde{d}, v)| \tag{2.13}$$

$$L_{tot} = L_W + L_{SSIM} + L_{Reg} \tag{2.14}$$

describing the image intensity difference between the matched points of the left and right image. As the disparity is not perfect some pixels will be moved too little or too much causing both empty gaps and positions with two contributions from the left image.

2.2.3.3 Linear interpolation

As the predicted disparity values $d(x, y)$ are float values and the positions must be integers the values must be rounded. To simply round the value to the nearest location can lead to loss of information. Instead of using the intensity value of the closest pixel an interpolation method can be applied which interpolates the intensity of pixels around the predicted float value. These values are weighted by the difference of the disparity float value and the integer real positions where respective intensity is taken from. For example, a position equal $[u + d_x, y] = [10 + 15.5, 20]$ would calculate the pixel intensity as $I_L(25, 20) * (25.5 - 25) + I_L(26, 20) * (26 - 25.5)$ for a linear interpolation. It can also be done for two variables such as the two coordinates (x,y) and is then called bilinear interpolation [26] which can be useful when both coordinates have float values.

Figure 2.3: Image warp example. To the left is the original right image and to the right is the warped image filled with the real right image according to the mask values.

2.2.3.4 Reconstruction mask

Other sources of warping-loss error are large texture-less regions and occluded regions. Texture-less regions are areas where the pixel intensities are so similar that it is difficult for a network to find matching features at the correct location. Occluded regions are areas only visible to one of the two cameras. As the network must see two pixels to perform a match this usually causes an error around the areas of edges to objects. A way to tackle the problem is to apply a mask that only give loss contributions from areas visible in both images and with enough texture to match pixels with. In that way the network is never punished for errors out of its control. By warping the left image and removing the pixels according to a calculated mask there will be blank spaces. If filled with the correct right image pixels the reconstruction task is eased of these tricky points. An example of how the combination of warped left image filled with true right image pixels can be seen in Figure 2.3.

2.2.3.5 Structural similarity between images

Image similarity loss is not always best compared with a MSE loss where a loss calculated with the SSIM values can be more accurate. The structural similarity index measure (SSIM) is a good complement to the MSE as it compares the similarity for a larger area and with a different method. It calculates the similarity of the luminance, contrast and structure between two images. The SSIM equation is as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.15}$$

where σ is the mean pixel intensity, μ is the variation and C_1, C_2 are constants to ensure computational stability. The mean and variation are calculated for each position a kernel is shifted over. The size of the kernel is one important parameter for the similarity measurement which decides how large area that is to be compared. As shifting all pixels one step could yield a large "warping-loss" from the MSE function the SSIM-loss will still be quite small since it focuses more on patterns and the overall kernel similarity [27].

2.2.3.6 Regularization loss

A way to increase network segmentation of the disparity map is to include a regularization loss in the loss-function. The regularization loss is defined as:

$$L_{Reg} = \frac{1}{N} \sum_{i \in N} (|\Delta_x^2 d_i| e^{-|\Delta_x^2 I_i|} + |\Delta_y^2 d_i| e^{-|\Delta_y^2 I_i|}) \tag{2.16}$$

where N is the number of pixels, d_i is the disparity value at position i and I_i is the image pixel intensity at position i. The disparity gradients $\Delta_x^2 d_i$ are weighted by the image gradients $\Delta_x^2 I_i$ such that sharp edges in both the image and disparity map do not result in a high loss. However, intermediate areas with low image gradients will give higher weight and hence large loss if the disparity gradient would be high in the same location. The effect is a smaller variation of disparities for connected areas such that an object normally is given less variation of disparity values.

2.2.4 IR night vision

Night vision is the ability to see in low-light conditions. One way to enable vision in dark environments is to use IR LEDs that emits electromagnetic radiation. IR light is defined as rays with wavelengths in the spectrum 700 nm to 1 mm [28] which are not visible to the human eye. Various kinds of IR LEDs produce light with different wavelengths and effects. By using a camera that can create images from reflected IR radiation night vision can be enabled. Two common IR night vision technologies that are used on the market today are far-infrared (FIR) and near-infrared (NIR) systems [29]. FIR cameras are used to detect thermal heat with the wavelength of around 8-12 μm. Warm objects will emit more radiation and will thus be more visible in the image. NIR cameras use near-infrared LEDs that emit radiation with a wavelength of 800 nm that the NIR camera can detect. The main advantage of NIR is the lower cost while FIR offer superior range.

2.2.5 Oriented FAST and Rotated BRIEF (ORB)

Oriented FAST and Rotated BRIEF (ORB) is an algorithm which describe and detect local features in images that was published in 2011 by Rublee et al. [30]. The algorithm is an efficient alternative to SIFT [31] and SURF [32] which is often used in computer vision applications like object recognition, image stitching and video tracking. ORB builds on the well-known FAST [33] key-point detector and the BRIEF [34] descriptor.

3
Methods

In this chapter the methods and performed experiments are presented. The hardware and software setups that were used during the project are first presented and elaborated on. Thereafter the data collection and network architectures are explained as well as the loss function. Lastly the methods used to evaluate performance and obtain results from are presented.

3.1 Hardware setup

In the project a single stereo camera rig was used to gather data and test the online performance with. The TX2 is built around a GPU NVIDIA Pascal™ with 256 NVIDIA CUDA® cores and a total RAM of 8 GB. The camera rig consists of four cameras mounted on a distance 14 cm apart. The cameras are from Leopard Imaging Inc [35]. They can capture images in full-HD at 60 fps, have a focal length of 5 mm and a pixel size of 3.75 µm. With the use of a cut-off filter the cameras can capture images in daylight with the filter and in total darkness without the filter if infrared (IR) light is used. Two 50 W IR LEDs are mounted to enable camera vision in the dark. The rig is equipped with one point laser rangefinder, LiDAR, that measure distances between 0 and 100 meters [36]. A photograph of the camera rig can be seen in Figure 3.1

Figure 3.1: Camera rig used in the project. From left to right are a camera, a 50 W IR LED, a camera, a laser rangefinder, a camera, another 50 W IR LED and a fourth camera. At the back, a Jetson TX2 is mounted and connected to the units.

In addition to the TX2 a stationary computer with a GeForce RTX3090 GPU [37] was used for training the neural networks. The GPU have 24 GB of G6X-memory and 10,496 CUDA cores.

3.2 Dataset Collection

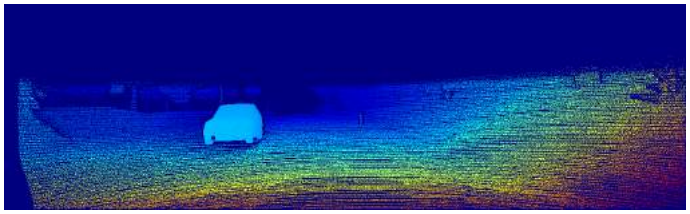
Training self-supervised neural networks require plenty of qualitative image data. The data need to consist of stereo image pairs from cameras with known baseline and focal length. In this project most training data were gathered with the camera rig in different environments. Furthermore, the famous KITTI stereo 2015 dataset was used.

3.2.1 KITTI Stereo 2015

KITTI vision benchmark suite is a project that was introduced by Karlsruhe Institute of Technology and Toyota Technological Institute in Chicago [38]. The purpose of the project is to offer challenging real-world computer vision benchmarks for different tasks like stereo disparity estimation, optical flow, visual odometry, 3D object detection and 3D tracking. The stereo dataset consists of rectified stereo image pairs with a resolution of approximately 1242x375 pixels. The dataset contains 8400 image pairs where only 200 of them have ground truth disparities. The ground truth data have been collected with LiDAR. The KITTI benchmark was used to verify that the proposed evaluation methods are accurate enough and to compare the network performance to other published networks. An example stereo image pair together with the ground truth disparity map can be seen in Figure 3.2.

(a) Left input image.

(b) Right input image.



(c) Ground truth disparity map.

Figure 3.2: (a) Left input image of stereo pair. (b) Right input image of stereo pair. (c) Ground truth disparity map.

3.2.2 Collecting new data

The stereo camera presented in Section 3.1 was mounted on moving vehicles and in static positions for different environments. Stereo calibrated images with a resolution of 1080x1920x3 were saved with a frequency of 1 FPS. The calibration was performed using Matlab's Stereo Camera Calibration App [39]. When a new dataset was collected the following tasks were performed:

1. Synchronized images were captured with the cameras
2. Images were stereo rectified with the calibration parameters
3. Images were labeled and saved on the disk

To test performance in different environments data were gathered in three ways. One dataset was gathered indoor to test the performance of an indoor light scenario. Another dataset was collected from a car with the camera mounted on top of the hover while driving around the city. To test the performance in low-light conditions one dataset was collected in a dark warehouse of 600 m^2. In this setup the cameras have a static position capturing scenes of moving objects and persons. Night vision was enabled during the gathering of data with use of the IR LEDs and cameras without cut-off filter.

3.2.3 Calibration

Calibration is necessary because of distortions and misalignment of the stereo camera. Distortions are created by the camera design since small angles and convexity of lenses affect the captured images. Small angular shifts between the stereo camera pair can also appear due to imperfect mounting. Offline stereo calibration compensates for most errors that are built into the camera rig. However, during usage the cameras are exposed to changes in temperature, pressure and outer forces which can cause physical changes that need to be compensated for. Then an online calibration strategy can become handy since it can be used to calibrate the cameras continuously during usage.

3.2.3.1 Offline stereo calibration

The cameras need to be calibrated to compensate for distortions as well as for angular and translational differences between the camera pairs. Offline calibration is performed by first calibrating the individual cameras separately and then stereo calibrate them together. During single camera calibration the intrinsic and extrinsic parameters are estimated individually for each camera. During stereo calibration the transformation (rotation and translation) between the two camera planes is estimated. The calibration was performed with the Matlab's Stereo Camera Calibrate App [39]. Multiple images of a chessboard with known dimensions were captured with all four cameras. The images were used as input to the calibration tool app whereby the intrinsic and extrinsic parameters were calculated. An example of an uncalibrated and calibrated image pair can be seen in Figure 3.3. The red channel represents the right image and the two other channels belong to the left image.

Figure 3.3: To the left is an uncalibrated image pair where red color channel represent the right image and the other channels represent the left image. To the right is the same image but calibrated.

3.2.3.2 Online stereo calibration

A way to avoid frequent offline calibration is by using automatic online calibration. By using the ORB algorithm corresponding key-points in the left and right images were found and the y-coordinates of the key-points were extracted. The difference between the y-coordinates of the two images were then used to determine how the images should be vertically shifted to match better. The following equation was used to decide the number of pixels to shift the images.

$$V_{offset} = \frac{1}{N} \sum_{i \in N} y_L(i) - y_R(i) \tag{3.1}$$

The left image was moved V_{offset} pixels to align better with the right image. This technique works well to horizontally align the images but unfortunately the disparities make it impossible to use the same technique to align the cameras with respect to the x-coordinate. An example of an online calibrated image can be seen in Figure 3.4. In addition, more advanced methods have proven capable of continuously estimating the stereo camera calibrations despite large initial errors and varying extrinsic parameters [40]. This is considered out of scope and will not be evaluated in the thesis.

3.3 Network architectures

Two different network architectures were investigated and implemented. The SH-Net inspired by GA-Net [11] and the XCNN [16] architecture that is a lightweight and simple architecture with promising performance. The networks were implemented in Tensorflow [41] and trained on benchmark datasets as well as on the data collected with the camera rig. A single loss function was developed and implemented for both networks such that the only difference was the network architecture.

Figure 3.4: Online self-adapting horizontal alignment using ORB. The red channel is from the right image while the green and blue channels come from the left image. To the left is the image pair before calibration, right image is shifted 15 pixels up. In the right image calibration has been performed and the image pair are horizontally aligned.

3.3.1 SH-Net

The Stacked Hourglass Network (SH-Net) was designed with inspiration from the feature extracting part of the GA-Net [11] that use a stacked hourglass architecture. The hourglass architecture consists of two down-sample parts with convolutional layers and two up-sample parts with transposed convolutional layers as can be seen in Figure 3.5. The down-sampling parts, called encoders, and the up-sampling parts, called decoders, compress the input to a latent space and then scale up to the input dimension again. To maintain information in the network residual connections, also known as skip-connections, are introduced between the encoders and decoders. These are represented in the figure as the yellow blocks and black arrows. This is a Siamese architecture which enables shared weights between the two identical networks and has proven effective for the stereo disparity estimation task [24]. There is also a rectified linear unit (ReLU) activation function in between every layer to normalize the values and ensure speed and stability in training. The last layer has a hyperbolic tangent activation function that set the output values between -1 and 1.

The two identical networks are fed with the left and right camera images with a width and height evenly dividable by 32 to keep consistent dimensions. The shared weights give a connection between the information of the left and right image such that, much like humans, the network can compare differences and estimate the disparity. The network can output one left and one right 2D disparity map with the same height and width as the input images. This network contains $4,336,658$ trainable parameters. A detailed table with all layers of the SH-Net can be found in Appendix A.

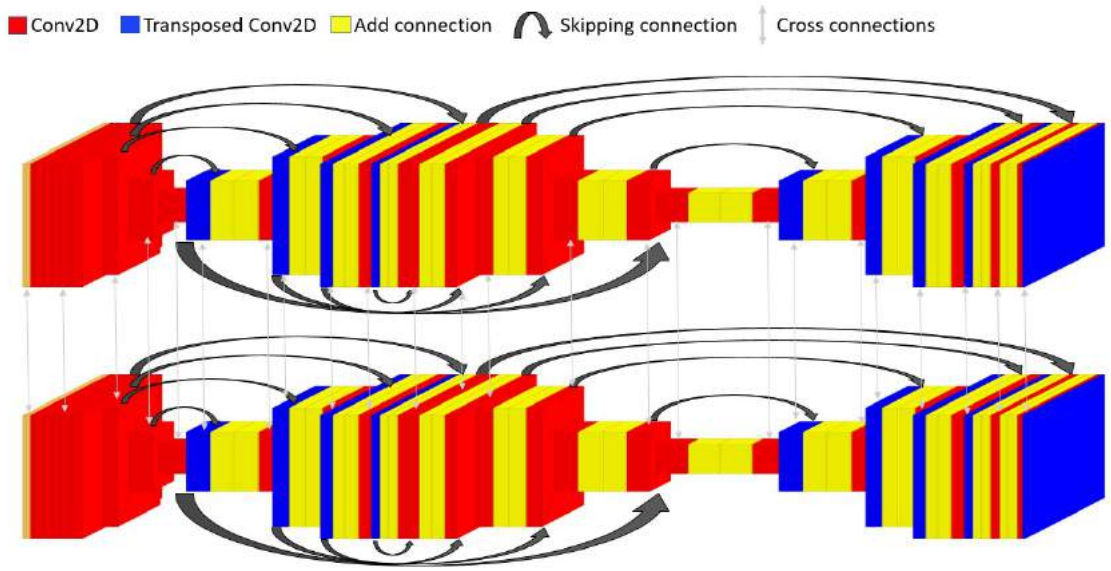


Figure 3.5: Siamese network architecture of the SH-Net with inspiration from GA-Net. Red layers are convolutional, blue are transposed convolutional and the yellow are adding layers connecting the encoder and decoder parts of the network. The two identical networks share weights through cross-connections (gray arrows) and have skip connections (black arrows) between the encoder and decoder parts.

3.3.2 XCNN

In earlier work another cross-connected network architecture was developed called XCNN [16]. Instead of the stacked hourglass architecture this network only has one encoder and one decoder but with similar structure and input/output dimensions. This network contains an encoder with 15 convolutional layers followed by a decoder with 11 transposed convolutional layers. The XCNN architecture can be seen in Figure 3.6. In difference to SH-Net this network has layers specific for the left and right network such that the weights are not shared for some of the layers. The outputs of those layers are instead combined such that contributions from respective side are added in a cross-connected way (black arrows in the figure). However, there are still shared weights for some of the layers and residual connections exist as well. With only one hourglass part and fewer channels for each layer this architecture became more light-weight then the SH-Net. The number of trainable parameters is less, 544,913, such that the amount of graphic memory is decreased. A detailed table with all layers of the XCNN can be found in Appendix B.

3.3.3 Loss function

To train the networks with the stereo images an unsupervised loss function was constructed. By minimizing this loss, the networks improved their ability to make disparity estimations. The predicted disparity map is used to warp the left image pixels horizontally to reconstruct the right image. If the disparity map is perfect

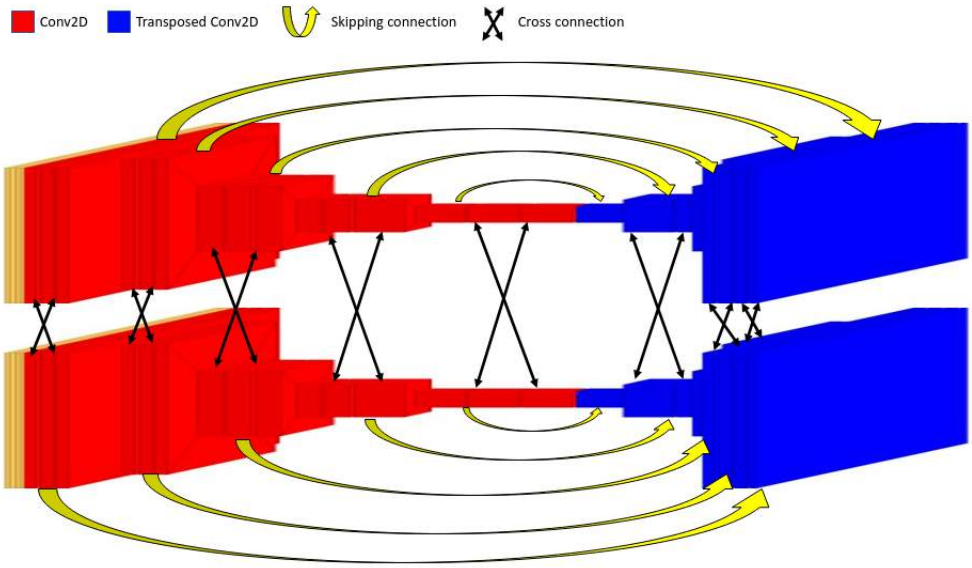


Figure 3.6: XCNN network architecture with skipping and cross connections visualized. The red layers are convolutional building up the encoder while the blue blocks represent the transposed convolutional layers that defines the decoder.

the warped left image equals the right image. However, an imperfect disparity map can be used for defining a loss. The imperfections of the disparity map cause a warped image with inaccurate pixel values. Such a warped image has gaps where no pixels were moved to and positions where multiple pixel values have been moved to. By using a reconstruction mask and bilinear sampling these imperfections could be handled. The gaps were filled with the nearest neighbor values and positions with multiple contributions were removed from the loss. The warped image and the right image were compared with respect to the pixel intensity difference. Both the raw difference from MSE and the more contextual SSIM difference. The warping loss was defined as:

$$L_{warp} = W_1 L_{MSE} + W_2 L_{SSIM} \tag{3.2}$$

W_1 and W_2 are weights that can be tuned to make the training more effective. A regularization loss, introduced in Equation 2.16, was added to the loss to make the disparity map more segmented. The regularization loss makes the disparity map smoother where the image gradient is small which results in a more segmented disparity map. The total loss was defined as a detail loss, L_{warp}, and a regularization loss, L_{reg}. Experiments show that a too high regularization loss can make the disparity map consistent for example, only filled with zeros [16]. On the other hand, without a regularization loss the image contrasts caused by details at the same distance from the camera give wrong disparity values. This causes an error in depth estimation and a balance in loss-weights must be found. The total loss function is expressed as:

$$L = W_1 L_{MSE} + W_2 L_{SSIM} + W_3 L_{reg} \tag{3.3}$$

The parameters that can be tuned are the three weights, W_1, W_2, W_3 and other specific loss parameters such as the kernel size of the SSIM filter.
Moreover, the loss function is necessary while training the networks but can be removed once the networks are fully trained. The prediction time during usage can then be lowered by removing the loss function of the trained networks.

3.3.4 FOV and optimized disparity zone

The horizontal field of view (FOV) is the open observable area that the cameras can see, and it can be calculated with:

$$FOV_{Horizontal} = 2arctan\left(\frac{width}{2f}\right) \tag{3.4}$$

The cameras had a width of 1920 pixels and focal length of $f = 1333.3$ [pixels] which gave a horizontal FOV equal 71.51^o. Given a baseline B this result in a closest common visible distance to the camera equal $Z = \tan(54.25^o) * B/2$. B is the baseline between the cameras and the angle is given by the geometry. For a baseline of 42 cm the closest common visible point is 29.3 cm from the cameras. Depending on the distance of an object there will be differently large *dark zones* at the edges of the disparity map. These zones are the areas only visible to one of the two cameras and hence difficult to obtain good disparities from. To increase computational speed and the percentage of accurate disparity estimations the edges can be cut off from both sides of the images. With the smallest depth measure of 3 meter the maximum disparity allowed was set to 189 [pixels]. This create *dark zones* of the same size at each side of the images which were cut off before input to the network. Furthermore, as interesting objects are usually located in the middle of the images the stereo image pair were cut with 200 pixels from the top and bottom, removing a lot of sky and road in the GBG traffic dataset. In Figure 3.7 the FOV for respective camera and possible occluded areas are presented. The idea was to cut of the *dark zones* that do not contribute to any qualitative information. In this way the need for memory and computational speed decreases. After trimming the images, they contain a size of $1542 \times 680 \times 3$, to fit the network dimensions had to be evenly dividable by 32. With this in mind, the true input size of the images was $1536 \times 672 \times 3$. By decreasing the image size in this way, the amount of input values was decreased from 6220800 pixels to 3096576 which is 50.2% less input values. Trimming the images was one way to increase the speed but performance was evaluated on full-HD images.

3.3.4.1 Optimal baseline

Theoretically, wider baselines yield better depth estimations for all visible ranges compared to shorter baselines. For example, if the network predicts a disparity map with one pixel offset it will yield larger distance errors for shorter baselines because of higher percentile errors compared to the same situation with larger baselines. Drawbacks of using a large baseline is that it yields larger occluded areas than short baselines and the minimum distance that can be estimated increases. In Figure 3.8 the disparity to depth relationship as well as the depth error caused by a pixel offset

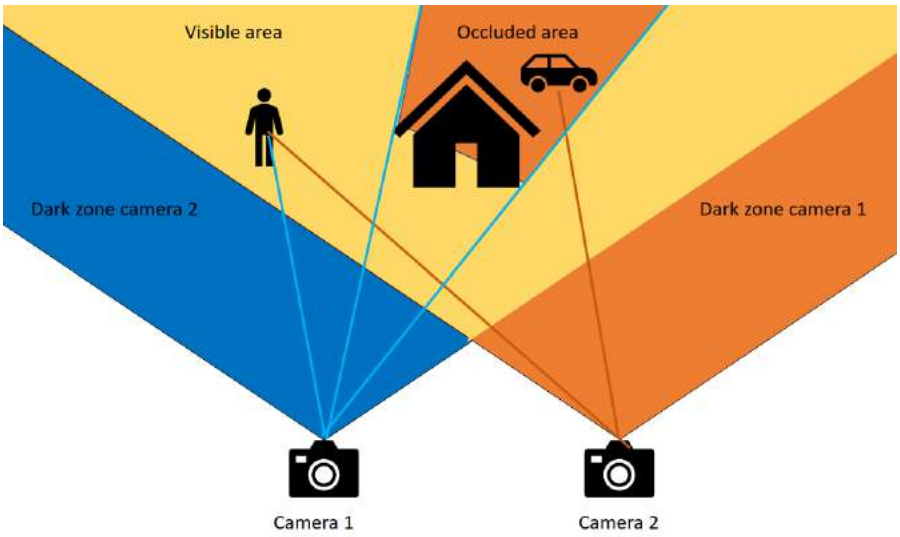


Figure 3.7: Field of view visualisation of visible and occluded areas for a stereo par. The *dark zones* at the edges as well as regions occluded by objects are marked with colors. Blue for areas not visible to camera 2 and the orange areas are not visible to camera 1. In this example there is a person visible to both cameras and a car only visible to camera 2 as the house occlude the car.

can be seen. The baselines are 14, 28 and 42 cm which are the possible baselines with the camera rig. The presented theoretical depth error is caused by a 3-pixel positive offset such that the error was calculated with the following equation,

$$E(d_L) = depth(d_L) - depth(d_L + 3) \tag{3.5}$$

where d_L is the disparity value and *depth* is a function converting disparity to depth measurement similar to Equation 2.11. The positive shift resulted in an error relative to the baseline such that larger baseline yield less error for the same disparity offset of 3 pixels.

To strengthen the theory an experiment was performed comparing the depth error from predicted disparities with two baselines of 14 cm and 42 cm. The camera rig was placed in a corridor gathering images with a person that walked in front of the camera back and forth in the corridor. More than 200 images were captured with depth information from the one-point laser. The stereo images were stereo rectified for the two baselines with the Matlab stereo calibration toolbox. Then the SH-Net was fine-tuned on the images with pre-trained weights that had more than 100 hours of training on the KITTI dataset with a baseline of 54 cm. After 10 epochs of fine tuning the depth error between the network and laser measurements were evaluated. The same procedure was applied for both baselines. In Figure 3.9 the left image and predicted disparity for the shorter baseline of 14 cm is presented. The predicted distance is 3.44 meter, the laser measured distance was 5.92 meter resulting in an error of -2.48 meter.

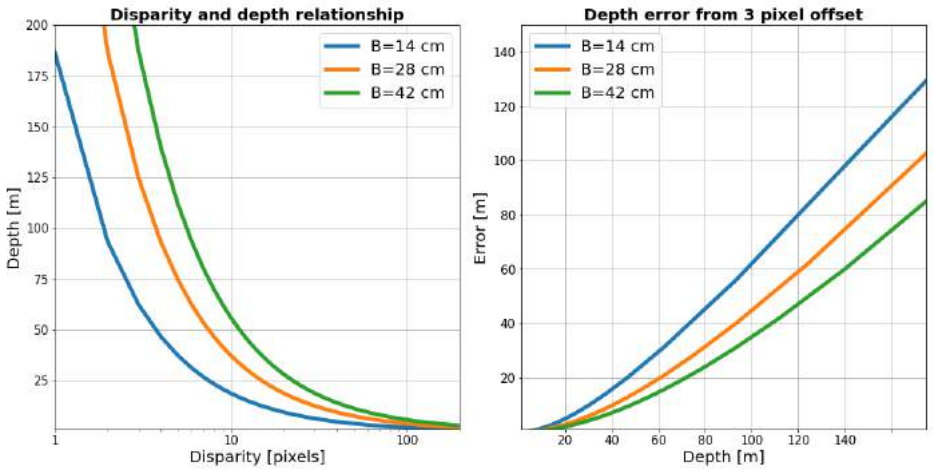


Figure 3.8: Disparity to depth relationship to the left and depth error caused by 3-pixel positive disparity error to the right. The lines show three different baselines that are possible with the given camera rig.

Figure 3.9: Depth estimation from stereo camera with baseline of 14 cm in an office corridor. The measurement value in red is the predicted distance, the laser measured 5.92 meter giving an error of -2.48 meters for this image.

3.3.5 Occlusion mask

One of the drawbacks of the stereo camera as a depth sensor is occluded regions. The offset between the cameras lead to areas that are visible in the left camera but not in the right and vice versa. It is difficult or even impossible for the network to reconstruct pixels from one image to another if it is not visible in both images. Consequently, calculating reconstruction loss on occluded pixels is noisy and will have a negative impact on network performance. Previous work show that an occlusion mask applied in the training result in less outliers for the predicted disparity map [16]. A solution that locates occluded pixel regions and exclude them from the calculated reconstruction loss. The occluded pixels were detected in the predicted disparity map through an iterative process. A pixel was classified as an occluded pixel if there existed another pixel in the left image that had been warped into the same coordinates in the right image. The network should be able to correctly warp all non-occluded pixels from the left image into the right. Furthermore, the network has problems predicting disparities in large textureless areas since no distinctive features can be extracted from these areas. False predictions in textureless areas will also cause the network to warp multiple pixels to the same pixel coordinates which will be highlighted by the occlusion mask.
In Figure 3.10, an example situation can be seen with an occlusion. In the top left image, the blue car is almost completely visible and in the top right image the rear of the blue car is not visible. The network fails to correctly reconstruct the left image by warping the right input image. The rear of the blue car cannot be reconstructed since the network was not able to match pixels in this area. Moreover, the *dark zones*, only visible to the left camera, were removed according to the occlusion mask as can be seen in the same figure (d)) where for example the left side was removed. The occlusion mask can also be used post training to refine predictions in occluded areas.

3.4 Evaluate depth accuracy

The depth measurement accuracy is one crucial evaluation parameter that was used to compare performance and robustness of proposed methods. The depth accuracy was evaluated in two ways. One method was to compare the measurements from the stereo camera with the on-board one-point laser range finder value, another method was to use the key-point matching algorithm ORB.

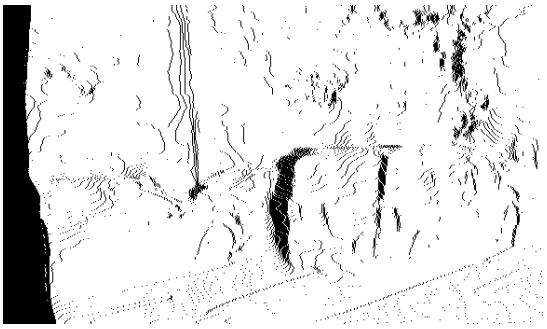
3.4.1 Depth evaluation using LiDAR

With the one-point laser range finder an accurate depth measurement was gathered with each stereo image pair. The on-board laser measurements have an accuracy of 0.1 m for a 70% reflective target at 20^0C. The laser was mounted in the middle of the camera rig pointing in the same direction as the four cameras. On top of a garage roof the camera was directed towards a small entrance where it captured images and laser measurements from different distances between 3 and 80 m. A left image, 20.84 meters from the stereo cameras, can be seen in Figure 3.11.

(a) Left input image.

(b) Right input image.

(c) Warped image from right to left.



(d) Occlusion mask.

Figure 3.10: The image has been warped using bilinear sampling based on disparity map predicted from left and right input images. The occlusion mask has highlighted occluded regions which can be seen as the black regions in the mask.

Figure 3.11: Left stereo camera image that depth was predicted from. On top of a garage roof with a distance of 20.84 m to the small entrance building according to the laser measurement.

The networks evaluated were the SH-Net and XCNN that were trained on the complete KITTI dataset and the custom created GBG traffic dataset. The networks were also fine-tuned on the evaluation images such that the specific environment on top of the garage was learnt. This training was done until no further improvements could be noted. Evaluation was performed such that the distance given from a laser measurement was compared with the network predicted distance for a point in the center of the garage entrance-building. The predicted distance measurement was the mean value of 200 pixels belonging to the object. The absolute difference of the predicted mean value and the laser point measurement was defined as the measurement error.

3.4.2 Depth evaluation using ORB

Since no ground truth data exists on the datasets collected it was difficult to evaluate network accuracy. One way to create ground truth data was to use ORB [30]. An algorithm was implemented in Python that took a stereo image pair as input and then returned pixel coordinates of matching features and their pixel disparities. The ORB algorithm was utilized to find pixel coordinates of matching key-points and descriptors. The ground truth disparity map was then computed by finding the horizontal pixel difference between key-points in the left and right image. Often, the algorithm produced a huge number of matching features but only a few matching key-points were accurate enough to compute ground truth disparities from. The algorithm sorted the matches in a list based on the certainty in ascending order with higher certainty in the front. The designed algorithm had an input parameter for the percentage of how many matching key-points that should be returned such that only good matches were obtained. A trade-off was made between accuracy and number of key-points when deciding the percentage of key-points to include as ground truth disparities. Different percentages were tested and it was decided that 2% of the found key-points were accurate enough to be used as ground truth pixels. The algorithm was evaluated to ensure that it could produce accurate ground truth values. This evaluation was performed on the KITTI benchmark dataset where the disparities from the ORB algorithm were compared with the corresponding LiDAR ground truth values. Approximately 100 matching key-points were found in each KITTI stereo image pair that could be used to compute disparities. However, the KITTI ground truth disparity maps are sparse and many of the computed disparities did not match a ground truth disparity value at that pixel coordinate. Due to the sparsity approximately 22 computed disparities per image could be evaluated against the ground truth data. In Figure 3.13a, 3.13b and 3.13c example image pairs from different datasets can be seen together with the key-points ORB has located in both images. The computed disparities that are evaluated against the ground truth are located randomly in the image which gave a good indication of the ORB algorithm performance. In Figure 3.12 the results can be seen from the evaluation, indicating that the mean absolute error was 0.77 px, variance 0.67 px and the standard deviation 0.82 px.

Even though ORB can estimate accurate disparities from stereo image pairs it is not suitable in a depth sensor application. The algorithm is too slow and the amount of

data that are retrieved per second are too sparse compared to the speed and amount of data retrieved from CNNs.

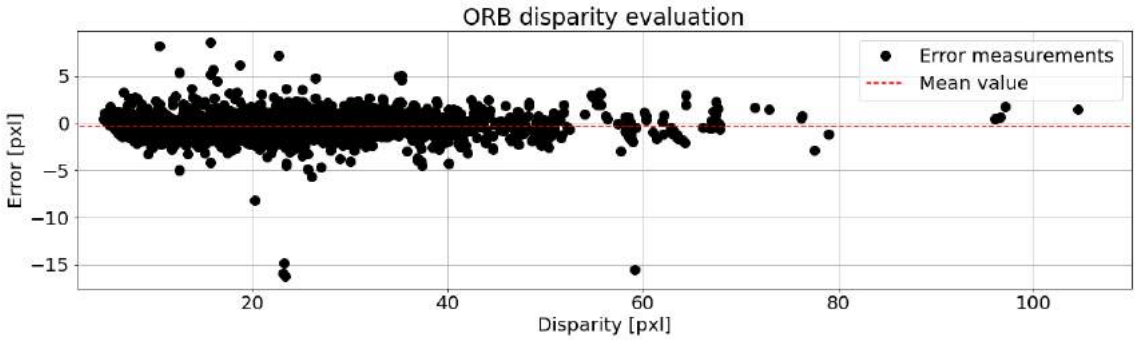


Figure 3.12: Error distribution of the ORB estimations compared with KITTI ground truth.

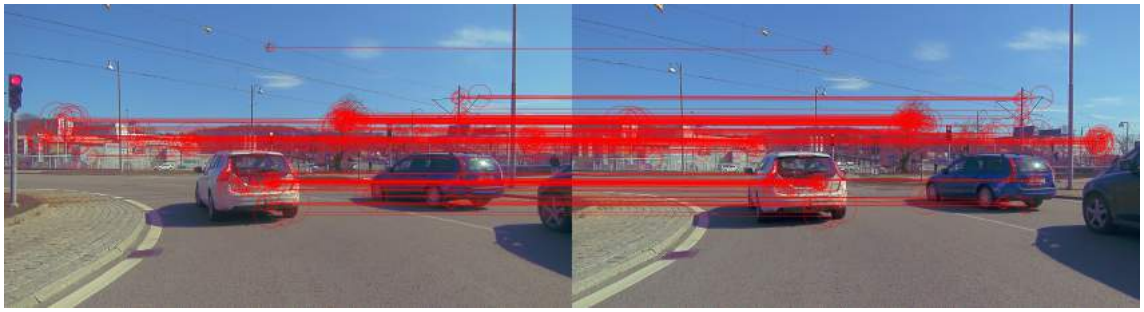
3.4.3 Depth evaluation in low-light conditions using IR light

For evaluation of the distance measurement with the cameras in low-light conditions a dark warehouse was visited. In the dark environment the camera rig with an IR LED of 50 W was aimed at an object that was moved between 2.5 and 22.5 meters from the cameras while capturing images. The distance to the object was also measured with the one-point laser such that ground truth values were saved. In Figure 3.14 an example image taken with one of the cameras in the dark can be seen. The networks used for depth evaluation were trained for more than 100 hours on the KITTI and GBG traffic dataset. The networks were also fine-tuned for 20 epochs on 550 image pairs captured in the same way as Figure 3.14 with moving objects in front of the camera.

3.5 Online self improving ability

With the self-supervised loss function that was used during training the networks can be self-improving and adapt themselves to new unseen environments. The stereo images taken by the camera can be used to predict a disparity map and simultaneously be used to calculate a reconstruction loss. The input images will serve as pseudo ground truth which will enable the network to fine-tune its weight parameters continuously during usage. An image from the GBG dataset was chosen to be the model of the adaptive improvement. The untrained SH-Net predicted a disparity map for the image, then it trained on 50 other images from the GBG dataset after which another prediction was made on the same image. The adaptive behavior is presented in Figure 4.7. The same procedure can be applied for a trained network with lower learning rate enabling a fine-tuning variant of adaption. While running the network in a self-improving mode the runtime is increased considerably since many operations are calculated in the loss function. In contrast, for a trained network without a loss function, only the operations of the network must be calculated which lower the computational cost during predictions.

(a)



(b)

(c)

Figure 3.13: (a) Image pair from Kitti benchmark dataset together with ORB key-points. (b) Image pair from GBG dataset together with ORB key-points. (c) Image pair from IR dataset together with ORB key-points.

Figure 3.14: Image captured with use of IR light in a dark warehouse.

4

Result and discussion

The results obtained from experiments and measurements described in the previous chapter will be presented and discussed here.

4.1 Optimal baseline

From the experiment presented in Section 3.3.4.1 the depth estimation accuracy of a stereo camera with baseline 14 cm and 42 cm can be compared. The errors from the depth predictions with SH-Net compared with ground truth laser measurements for the two baselines are presented in Figure 4.1. The mean error is closer to zero for the larger baseline and for short distances the maximum error is larger for a baseline of 42 cm compared with 14 cm. On the other hand, for long distances the opposite applies such that the shorter baseline gives larger error outliers. In Figure 4.2, the distribution of the errors calculated from the two baselines are presented. Increasing the baseline from 14 cm to 42 cm seem to decrease both the mean error and the error variation. With 14 cm between the cameras the mean error was -3.20 meter and the standard deviation was 1.84 meter. Increasing the baseline to 42 cm results in a mean error of -0.47 meter and a standard deviation of 1.74 meter. These results indicate that for distances between 3 and 22.5 meter a baseline of 42 cm have better performance than a baseline of 14 cm.

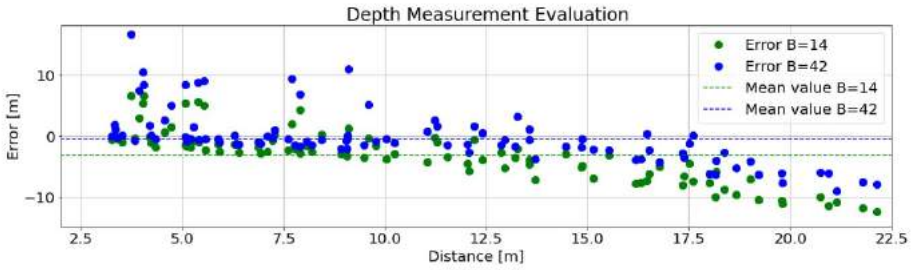


Figure 4.1: Depth errors calculated for the two baselines of 14 and 42 cm. The measured errors are plotted together with the mean value for both baselines at distances between 2.5 and 22.5 meter.

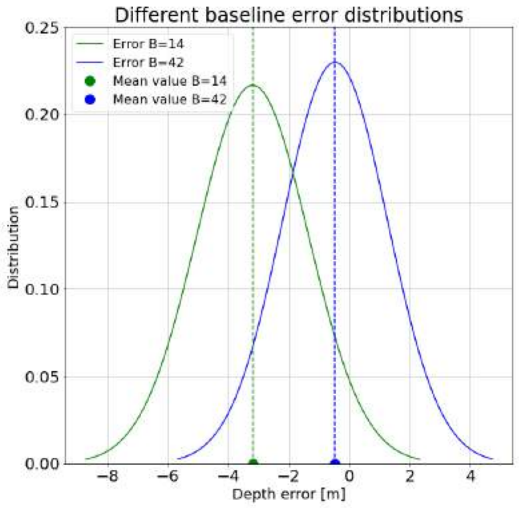


Figure 4.2: Distribution of the measured errors for the two baselines of 14 and 42 cm. The error measured with baseline equal 42 cm is closer to zero and less spread.

4.2 Numerical results

In this section the numerical results from the evaluation of the stereo camera as a depth sensor are presented. The evaluation has been performed using the software Keras [42] with a TensorFlow backend [41]. Two networks, SH-Net and XCNN, have been implemented and evaluated separately. The performance of the networks was first evaluated on KITTI 2015 benchmark to compare performance with state-of-art network architectures. Thereafter the real-life depth estimation performance of the networks combined with the binocular camera were evaluated. The real-life performance was evaluated by training and evaluating the networks on self-collected data captured with the binocular camera presented in Section 3.1. Ground truth data were created by measuring distances to objects with a one-point laser range finder and by using the ORB algorithm presented in Section 3.4.2.

4.2.1 KITTI benchmark

The two networks implemented were evaluated on the KITTI 2015 benchmark dataset. The dataset contains 200 evaluation images with ground truth data that has not been seen during training. The performance was evaluated by calculating "D1-all" which represents the percentage of outliers averaged over all the ground truth pixels of the 200 test images. A pixel was classified as an outlier if the disparity was falsely predicted with more than 3 pixels. The results from the evaluation can be seen in Table 4.1 and are compared to LWA-Net which is one of the most recent state-of-art self-supervised network architectures with an impressive runtime. The runtimes have been calculated for predictions on the Jetson TX2 with 256 CUDA cores. What can be observed from the results is that LWA-Net is the best performing network on this benchmark. XCNN has better accuracy than SH-Net but is the slowest one.

Table 4.1: Evaluation results on Kitti 2015 benchmark for different self-supervised network architectures. Unavailable data are noted as -.

Network	Parameters (million)	D1-all (%)	Average Runtime (s)	RMSE	Input size (pxl)
SH-Net	4.336	18.7	0.45	7.33	320x1216x3
XCNN	0.545	7.711	0.50	3.6	320x1216x3
LWA-Net	0.098	4.94	0.20	-	320x1216x3

4.2.2 Depth sensor evaluation

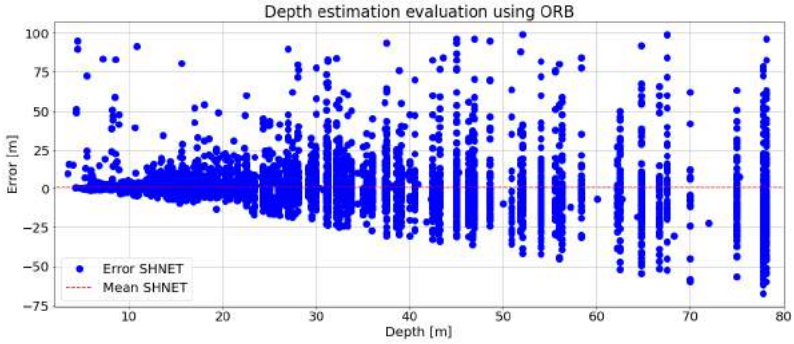
The networks were trained on a dataset collected with the outermost cameras with baseline 42 cm on the stereo camera rig described in Section 3.1. The evaluation was performed both in daylight and in low-light conditions where the datasets were collected as described in Section 3.2.

4.2.2.1 Depth evaluation using ORB

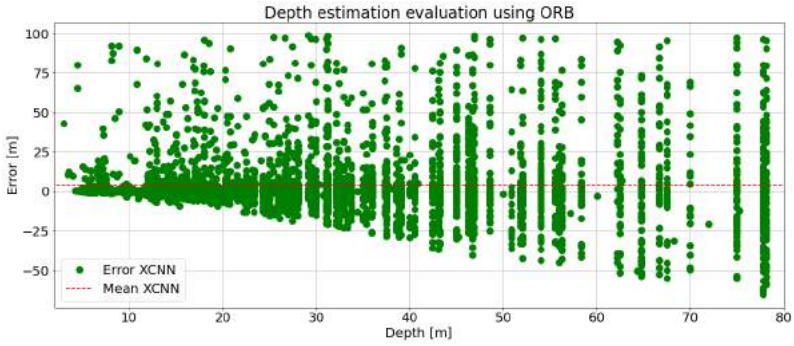
The depth sensor was evaluated using ground truth depth computed with ORB. The evaluation dataset contains 40 images captured during daylight that has not been seen during training. Approximately 130 ground truth depth pixels were computed per image in the evaluation dataset. To make a thorough evaluation of the networks and understand their strengths and weaknesses the evaluation has been performed on different distance intervals. During evaluation it was noted that sometimes the networks made false predictions with several hundred meters for distances longer than 80 meters. Therefore, the decision was made to not include predictions longer than 80 meter in the result. The evaluation result for XCNN and SH-Net can be seen in Table 4.2. The average runtimes are calculated on a GeForce RTX3090 with 10,496 CUDA cores and on the Jetson TX2 with 256 CUDA cores. In Figure 4.3a and 4.3b the error measurements are plotted for both networks. The depth absolute mean error and standard deviation of SH-Net were 10.25 and 13.94 meter respectively. The depth absolute mean error and standard deviation of XCNN were 11.37 and 17.9 meter. What can be noted from the plots and in the table is that the prediction error increases with the distance which is not the case for the disparity error that seems to decrease with the distance. Both networks have the best accuracy for distances from 3 to 20 meter where SH-Net is slightly better.

4.2.2.2 Laser point evaluation

Table 4.3 presents the real and relative distance errors between predicted values and measured distances with a laser rangefinder. The overall spread of errors for this result is presented in Figure 4.4. For short distances the prediction consistently overshot and for the distances longer than 20 meter the predicted distances mostly undershot. The SH-Net has less outliers while XCNN tend to predict too long distances and spread out the predictions more, especially for the longer distances.



(a)



(b)

Figure 4.3: (a) Error distribution of predictions from SH-Net on GBG dataset. (b) Error distribution of predictions from XCNN on GBG dataset.

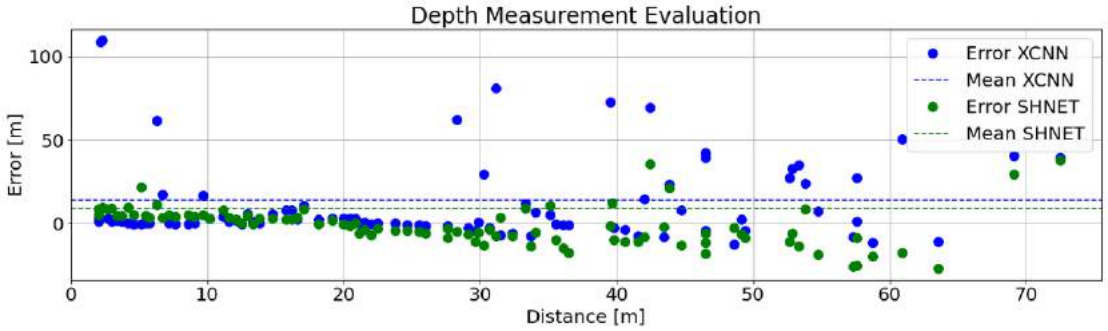


Figure 4.4: Error depth calculated between predicted disparity estimation and laser measurement. Images of a garage entrance building at distances between 3 and 73 meters were input to the SH-Net and XCNN network to estimate the depth.

Table 4.2: Depth and disparity evaluation result from the two network predictions compared with ground truth from the ORB algorithm on GBG dataset.

Network	**SH-Net**	**XCNN**
Input size [pxl]	1920x1056x3	1920x1056x3
Runtime on GeForce RTX3090 [s]	0.048	0.055
Runtime on Jetson TX2 [s]	1.78	2.24
Baseline [cm]	42	42
Disparity absolute mean error all [pxl]	6.73	7.61
Disparity absolute error standard deviation all [pxl]	9.55	13.5
Disparity absolute error mean 3 -> 20 meter [pxl]	9.73	9.31
Disparity absolute error mean 20 -> 40 meter [pxl]	6	5.53
Disparity absolute error mean 40 -> 60 meter [pxl]	4.87	5.37
Disparity absolute error mean 60 -> 80 meter [pxl]	4.12	4.12
Depth absolute mean error all [m]	10.25	11.37
Depth absolute error standard deviation all [m]	13.94	17.94
Depth absolute error mean 3 -> 20 meter [m]	3.26	4.09
Depth absolute error mean 20 -> 40 meter [m]	8.45	9.32
Depth absolute error mean 40 -> 60 meter [m]	13.39	15.57
Depth absolute error mean 60 -> 80 meter [m]	18.5	21.98

Network	Error mean	Error std	Relative error mean	Relative error std
SH-Net	8.79	3.51	62.80%	92.51%
XCNN	13.56	4.24	150.15%	152.36%

Table 4.3: Depth error of network predictions from images captured in daylight on top of a garage roof. The error is defined from the predicted depth compared with laser measurements for distances between 3 to 73 meters.

4.2.2.3 Performance in low light conditions using laser

The performance of the networks in low light conditions were evaluated in the same way as the laser point measurement evaluation but in a dark warehouse and with use of IR light. The resulting error distribution of distances between 2.5 and 22.5 meter can be seen in Table 4.4 and Figure 4.5. The network tends to estimate longer distances than what the actual distances were. The SH-Net has a more consistent error with uncertainty that grow with the distance. The mean error was 0.974 meter in average, the standard deviation was 0.820 meter and the largest error was 4.31 meter. The XCNN was more inconsistent and had some predictions with larger errors. The mean error was 3.649 meter and standard deviation was 2.152 meter where 6 of the 65 predictions have the largest contribution. The largest error was 36.12 meter but except for the 6 outliers most predictions were as accurate for the XCNN as they were for SH-Net. Apart from the outliers the XCNN depth errors are more spread at longer distances but the relative error decreases as the distance grow.

Network	Distances	Error mean	Error std	Relative error mean	Relative error std
SHNET	2 < x <22.5	0.974	0.820	10.03%	25.82%
XCNN	2 < x <22.5	3.649	2.152	25.63%	51.63%

Table 4.4: Depth error from SH-Net and XCNN network prediction compared with laser measurements for distances between 2.5 to 22.5 m. Images were captured in a dark warehouse with IR lights as the only source of light.

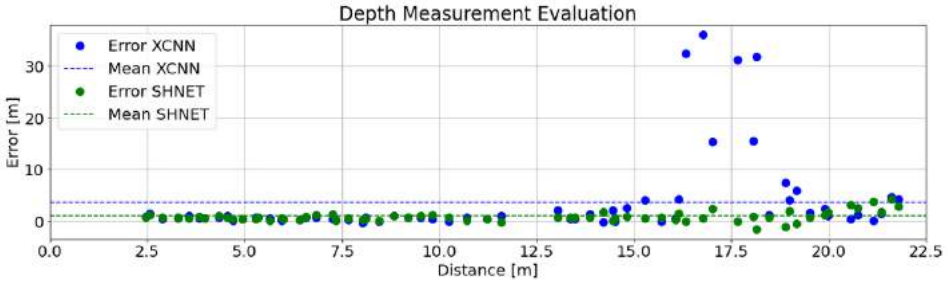


Figure 4.5: Depth errors from 65 image pairs collected in the dark with IR light as only source of light. Distances between 2.5 and 22.5 meter were measured and the error for the two networks, XCNN and SH-Net, were calculated as the difference of predicted depth and measured depth using laser.

4.2.2.4 Performance in low light conditions using ORB

The performance of the networks was further evaluated in the dark warehouse using ORB. Approximately 55 ground truth depth pixels were computed per image for this evaluation dataset. The result from the evaluation of XCNN and SH-Net can be seen in Table 4.5. In Figure 4.6 the error measurements are plotted for both networks. The depth absolute mean error and standard deviation of SH-Net were 2.49 meter and 5.43 meter. The depth absolute mean error and standard deviation of XCNN were 4.56 meter and 10.87 meter. What can be interpreted from the table and plots is the same behavior as in daylight. The depth error increases with distance while the disparity error decreases. It can also be noted that the depth prediction accuracy seems to be better for images captured in IR light compared to daylight.

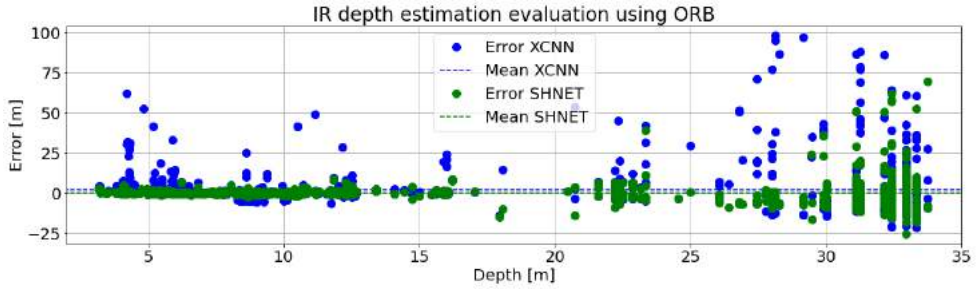


Figure 4.6: IR depth estimation errors for distance 0-35 meter for IR dataset.

Table 4.5: Depth and disparity evaluation result from the two network predictions compared with ground truth from the ORB algorithm on IR dataset.

Network	SH-Net	XCNN
Input size [px]	1920x1056x3	1920x1056x3
Baseline [cm]	42	42
Disparity absolute mean error all [px]	5.46	9.35
Disparity absolute error standard deviation all [px]	8.03	17.65
Disparity absolute error mean 3 -> 20 meter [px]	5.85	10.71
Disparity absolute error mean 20 -> 40 meter [px]	4.65	6.13
Depth absolute mean error all[m]	2.49	4.56
Depth absolute error standard deviation all [m]	5.43	10.87
Depth absolute error mean 3 -> 20 meter [m]	0.6	1.57
Depth absolute error mean 20 -> 40 meter [m]	6.48	11.61

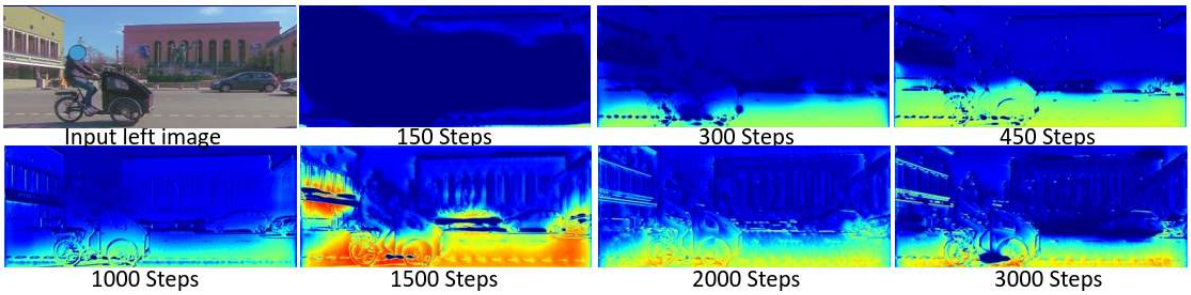


Figure 4.7: Example of SH-Net adapting to data through training on the GBG traffic dataset. The network start with untrained weights and improve during 3000 training steps.

4.3 Adaptive performance

The networks can adapt to different environments by continuously updating the weight parameters during usage as depth sensor. An improvement of the untrained SH-Net while training on the GBG traffic dataset is presented in Figure 4.7. The improvements are distinct for the first 450 training steps and then learning is slower. For some steps there is no improvement or even a deterioration for the prediction result. In this setup the learning rate was $lr = 0.0005$ and the training weights introduced in Equation 3.3 were set to $W_1 = 0.5$, $W_2 = 0.8$ and $W_3 = 0.3$. Furthermore, fine-tuning already trained weights was also done during the project and with a balanced learning rate the network can adapt both fast and correct to new scenes.

4.4 Visual results

A visual evaluation of predicted disparity maps was performed to better understand the strengths and weaknesses of the networks.

4.4.1 GBG traffic dataset

In Figure 4.8b the best SH-Net prediction result from the GBG evaluation dataset can be seen. The result of this prediction had a depth relative mean error of 10.3% and depth error mean of 1.32 m. When observing the predicted disparity map the prediction looks accurate where the car, trees, buildings and other relevant objects have been given reasonable disparity values. In Figure 4.8d the worst SH-Net prediction result from the GBG dataset can be seen. The result from this prediction had a depth mean relative error of 137% and depth error mean of 28.65 m. When observing the predicted disparity map, it seems like the network has captured the car correctly but had problems with the fence and the houses in the background. The horizontal subway wires also caused too high disparities in the sky leading to large distance errors.

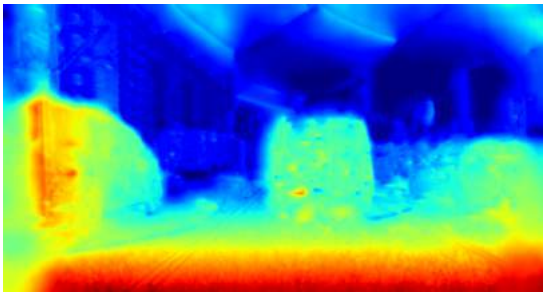
In Figure 4.9b the best prediction result from XCNN can be seen from the GBG traffic dataset. The result of this prediction had a depth relative mean error of 12.9% and depth error mean of 1.85 m. When observing the predicted disparity map it seems like the network captured the car in the middle correctly but had some problems with the left and right car as well as the road. In Figure 4.9d the worst prediction result can be seen from XCNN from the GBG dataset. The result from this prediction had a depth relative mean error of 111% and depth error mean of 36.6 meter. When observing the predicted disparity map, it seems like the image did not contain any distinct object and the network had problems predicting the road and the tress correctly.

4.4.2 IR dataset

In Figure 4.10b the best prediction result from SH-Net can be seen from the IR evaluation dataset. The result of this prediction had a depth relative mean error 4.7% and depth error mean is 0.37 m. When observing the predicted disparity map the prediction looks accurate, both persons and the trash bins were captured by the network. In Figure 4.10d the worst prediction result can be seen from the IR dataset. The result from this prediction had a mean relative error of 22% and depth error mean of 6.75 m. When observing the predicted disparity map it seems like the prediction is accurate, but the network had some problems with the area in the lower left corner.

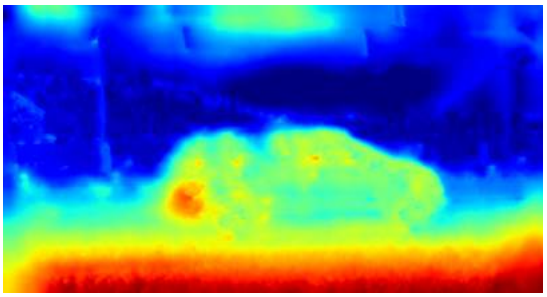
In Figure 4.11b the best prediction result from XCNN can be seen from the IR evaluation dataset. The result of this prediction had a depth relative mean error of 5.9% and depth error mean of 0.32 m. When observing the predicted disparity map the network has captured the persons and objects correctly. The contours of the objects are sharper compared with the prediction from SH-Net. In Figure 4.11d the worst XCNN prediction result for the IR dataset can be seen. The result from this prediction had a depth mean relative error of 53% and depth mean error of 12.79 meter. When observing the predicted disparity map, it seems like the network have captured the objects correctly but had some problems with the background.

(a) Left input image of best prediction.



(b) Best predicted disparity map.

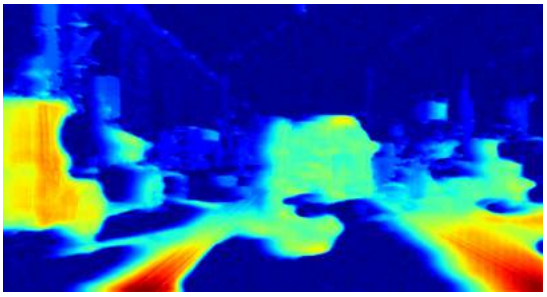
(c) Left input image of worst prediction.



(d) Worst predicted disparity map.

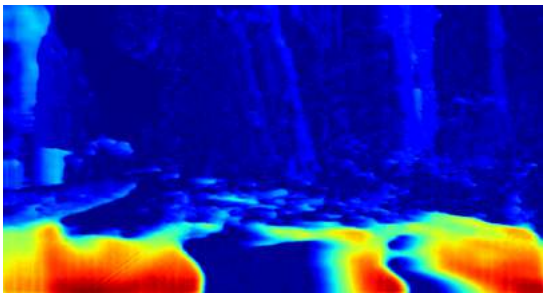
Figure 4.8: (a) Left input image that produced best prediction result from SH-Net. (b) Best predicted disparity map from SH-Net, relative error mean 10.3%. (c) Left input image that produced worst prediction result from SH-Net. (d) Worst predicted disparity map from SH-Net, relative error mean 137%.

(a) Left input image of best prediction.



(b) Best predicted disparity map.

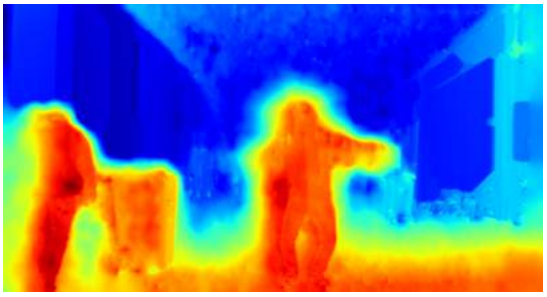
(c) Left input image of worst prediction.



(d) Worst predicted disparity map.

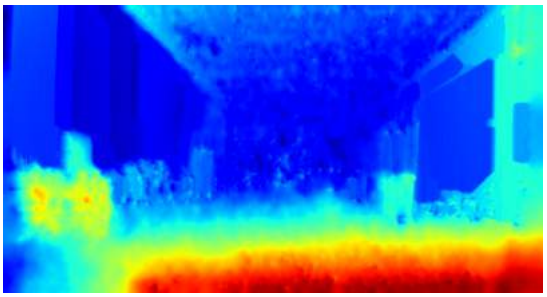
Figure 4.9: (a) Left input image that produced best prediction result from XCNN. (b) Best predicted disparity map from XCNN, relative error mean 12.9%. (c) Left input image that produced worst prediction result from XCNN. (d) Worst predicted disparity map from XCNN, relative mean error 111%.

(a) Left input image.



(b) Best predicted disparity map.

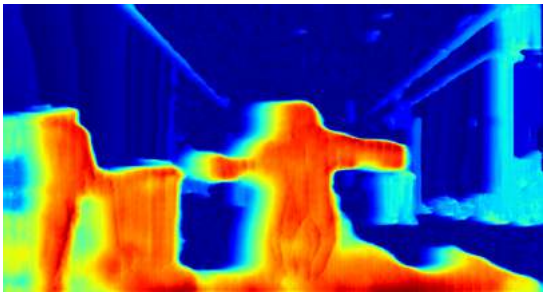
(c) Left input image.



(d) Worst predicted disparity map.

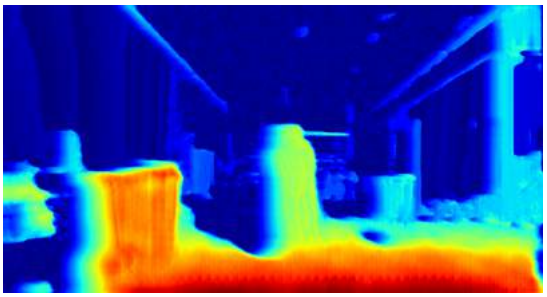
Figure 4.10: (a) Left input image that produced best prediction result from SH-Net. (b) Best predicted disparity map from SH-Net, relative error mean 4.7%. (c) Left input image that produced worst prediction result from SH-Net. (d) Worst predicted disparity map from SH-Net, relative error mean 22%.

(a) Left input image.



(b) Predicted disparity map.

(c) Left input image.



(d) Predicted disparity map.

Figure 4.11: (a) Left input image that produced best prediction result from XCNN. (b) Best predicted disparity map from XCNN, relative error mean 5.9 %. (c) Left input image that produced worst prediction result from SH-Net. (d) Worst predicted disparity map from SH-Net, relative error mean 53 %.

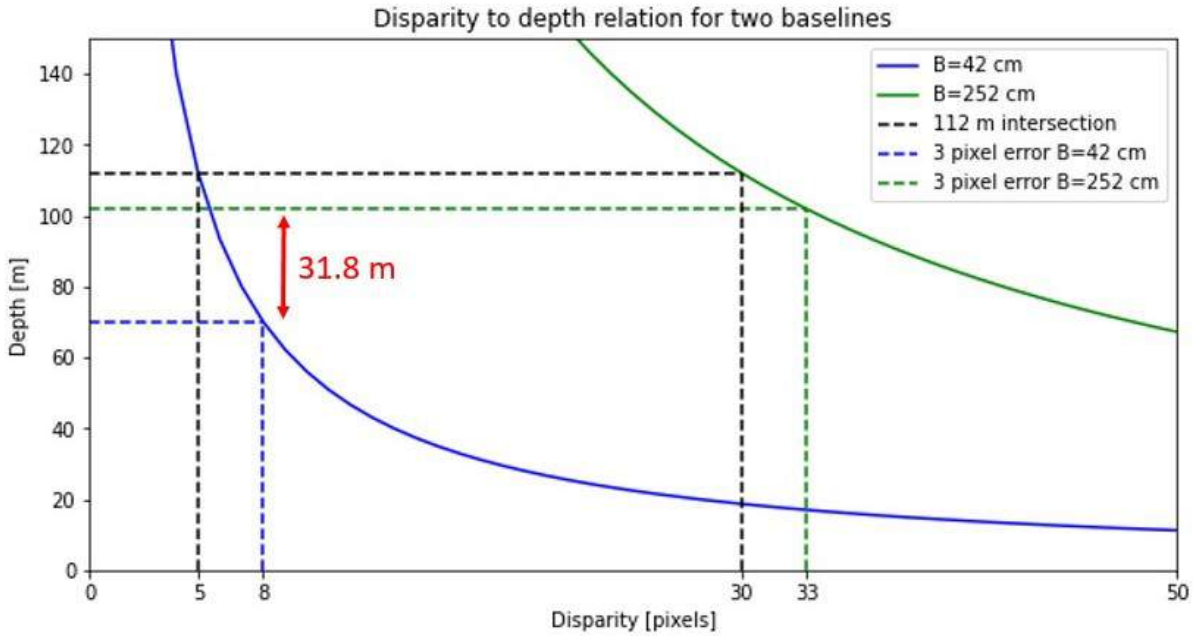


Figure 4.12: Disparity to depth relation for two baselines with the distances 42 cm and 252 cm. The corresponding disparity for 112 meter measurements are plotted together with a 3-pixel positive offset. The offset causes an error in depth measurement with more effect on the shorter baseline.

4.5 Discussion

In this section a discussion of the presented result will take place. With a focus on why the result look like it does this part will connect the result and conclusion.

4.5.1 Optimal baseline

Previous work with stereo camera concludes that a short baseline is better at predicting depth at short distances while a larger baseline increase performance at longer distances [43]. However, the theory and experiments performed in this work indicate the opposite. With respect to depth measurement accuracy a wider baseline is always more accurate and robust compared to a smaller baseline setup. The only drawback of using wider baselines is the increased number of occluded regions and that the shortest measurable distance increases. Instead of evaluating baselines less than 42 cm it would be interesting to learn the accuracy and robustness of baselines up to several meters. The relation between disparity and depth, for baselines with distances 42 cm and 252 cm, is presented in Figure 4.12. A positive 3-pixel offset is plotted for the baselines as well as the resulting depth for such an error. At 112 meters the predicted depth is 31.8 meters more wrong for the 42 cm baseline compared with the 252 cm baseline. A distance of 112 meters is therefore estimated more robustly with the wider baseline. The result from this thesis indicates that distances from 40 meter and more are difficult to predict reliable disparity maps for with the current maximum baseline of 42 cm. However, as the performance between

3 to 20 meter was much better, increasing the baseline should make the disparity prediction more accurate and robust. For example, if a baseline of 252 cm would be used this should give the same accuracy for distances from 18 to 120 meters as the baseline of 42 cm gives for 3 to 20 meters. It should be possible to setup three cameras such that one long and one short baseline are obtained and thus both short and long distances can be measured more accurate and robust.

4.5.2 Quality of data

The quality of the training data is very important. If the data is ill calibrated, have too little variance or other disturbances it will be more difficult for the networks to learn disparity estimations. The quality of the KITTI dataset is probably better than the data collected with the camera rig since it was a larger project with the only focus of creating qualitative data. When gathering the GBG dataset small errors can have corrupted the data. In that case it was most probably due to disturbances during the capturing of stereo pairs or due to an imperfect camera calibration and image rectification. If the images were not captured in perfect synchronization objects might move during the time difference of capturing the right and left image. Then the disparity map is no longer valid for the objects that moved. Likewise, if any calibration parameter is wrong or the rectification is not perfect the disparity input to the networks will be corrupted leading to bad predictions. It will also affect the disparity to depth relationship since the focal length was calculated in the calibration.

4.5.3 Network performance

The networks used in this project do not perform well on the KITTI benchmark dataset compared with other networks. Even though the benchmark dataset performance give an indication of how qualitative the disparity estimations are it does not entail that the general depth measurements are equally good. Presented experiments indicate that the quality of network depth predictions do not always align with the performance on benchmark datasets. The XCNN network was much better than SH-Net on the KITTI benchmark but for the depth measurements on data collected with the camera the SH-Net had better performance. There are many possible reasons why this is the case but since the loss-function, training parameters and data are the same it is probably the network architecture that have the largest impact. The two networks are very different and with respect to trainable parameters the SH-Net is more than seven times larger. It is possible that SH-Net can learn more complex features and patterns since there are more parameters to change. However, the network could learn a too complex pattern such that unimportant details in images have more effect on the disparity map. A less segmented and more noisy disparity map are two of the possible negative consequences of this. XCNN have layers that do not share weights between the parallel lanes, and this could affect what intermediate features that are obtained. For example, an occluded region only visible to one of the cameras could be processed in two different ways in the layers that do not share weights in XCNN. This would be more complicated for SH-Net

since all weights are shared and the left and right input image will therefore be processed in more similar ways.
The relation between disparity and depth has a large impact on the performance and the distance to objects in an image will therefore impact the accuracy that depth is measured with. Another parameter affecting the quality of measurements are the cameras. Imperfect lenses, sensors and calibration can impact how the images are captured such that assumptions regarding focal length, pixel size, baseline and horizontal alignment become invalid. With use of ground truth values obtained with laser measurements and the ORB algorithm the accuracy of depth measurements can be evaluated. By evaluating the depth rather than the disparity the true performance of a stereo camera as a sensor can be evaluated. Most published projects only measure the disparity accuracy for a given benchmark dataset. In order to obtain information about the performance of a stereo camera as depth sensor we propose that also the depth accuracy is evaluated. With use of LiDAR or matching algorithms the performance can be evaluated numerically.

4.5.4 Adaptive performance

An adapting behavior can be beneficial for the network if the general performance can be preserved. We show that it is possible to implement an adaptive depth sensor, through self-supervised learning, but not how well such a system would perform in real situations. In order to be useful, the time to adapt and quality obtained after such an adaption must be evaluated further. The obtained result gives an indication that adaptive training is both effective and useful for new unseen environments.

4.5.5 IR depth accuracy

Surprisingly the network performance with IR light in darkness was better than outdoor in daylight. Exactly why this is the case is difficult to find out but one major difference between the input data are the color channel intensities. In daylight images the color channels intensities are more similar. For the IR images the green color channel only contain low intensities while the blue and red channels are more centered. An example of the color spectra for one daylight and one IR light image can be seen in Figure 4.13. Another mayor difference is that for IR images the camera rig was placed in a static location such that it could fit the network to the environment during training. The daylight images from GBG traffic dataset are all taken from different positions such that the background change on all images while training. These two differences are the two most probable reasons for the result obtained.

4.5.6 Visual performance in daylight

When comparing the visual performance of SH-Net and XCNN the SH-Net seems to give better estimations of the background compared to XCNN. In Figure 4.14b the worst prediction from XCNN can be seen compared to the prediction from SH-Net in Figure 4.14c. What can be observed is that XCNN have trouble predicting the road and the forest on the right side of the road. This is not the case for SH-Net which

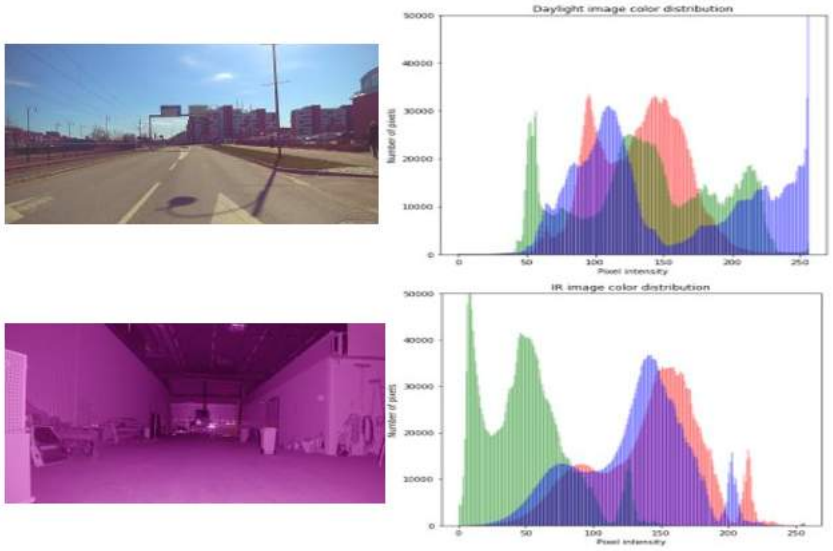
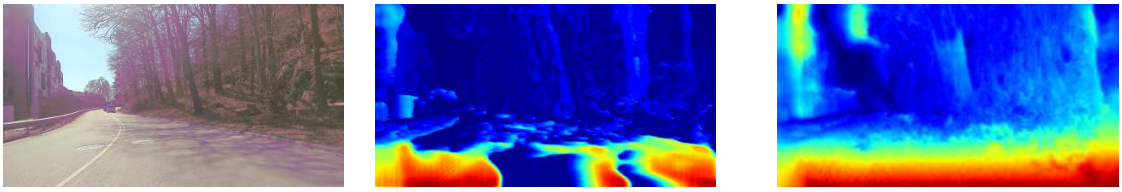


Figure 4.13: Color distribution for red, green and blue channels taken from two example images. One image taken in daylight on a road and another image in a dark warehouse with IR as light source.



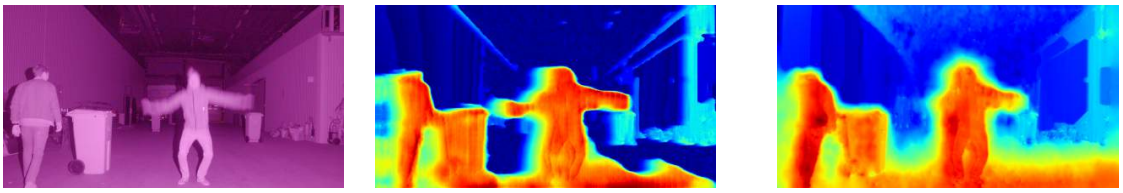
(a) Left input image. (b) XCNN prediction. (c) SH-Net prediction.

Figure 4.14: Prediction comparison between XCNN and SH-Net for input image without distinctive objects.

has captured the road and the trees correctly. Why SH-Net performs better than XCNN for these kind of input images without distinctive objects is probably because the SH-Net can learn more complex features due to its larger network architecture.

4.5.7 Visual performance with IR light

When comparing the visual performance of the networks in IR light both networks seem to have the ability to detect objects from 3 to 35 meter accurately. When comparing the predictions in Figure 4.15 it can be observed that the XCNN seem to have captured the objects more detailed than SH-Net. When observing the prediction of the person in the middle of the image XCNN have captured both arms correctly while SH-Net has problems with the left arm. It also seems that the XCNN handles occluded regions better since the objects have more sharp edges compared to SH-Net predictions. It is hard to give an exact reason why the XCNN handles occluded regions better than SH-Net. The most probable reason is the differences in the network architecture where the XCNN has layers that are not cross connected



(a) Left input image. (b) XCNN prediction. (c) SH-Net prediction.

Figure 4.15: Prediction comparison between XCNN and SH-Net for IR input image.

which may improve predictions of pixels that are only visible in one of the cameras.

4.5.8 Evaluation using LiDAR and ORB

The biggest challenge of developing an accurate depth sensor using a CNN and a stereo camera is to ensure the quality of the depth estimations. Even though the CNN can be trained using self-supervised learning the network performance needs to be evaluated on that specific camera configuration. Furthermore, to ensure quality of the depth sensor ground truth data is needed. Collecting accurate ground truth data can be laborsome and expensive which can be a major drawback of developing a stereo camera depth sensor. How extensive the evaluation ground truth data needs to be depends on the application and requirements of the depth sensor. If the stereo camera is to be used in safety critical applications, the ground truth data need to be dense and accurate which might be less important in for example a surveillance application.
LiDAR and ORB is two proposed methods that can be used to design ground truth for evaluation. The ground truth data created with these methods are accurate but sparse. The laser range finder will produce an accurate measurement, but it requires manual work to find which area in the image that is represented by the laser. Using key-point matching algorithms like ORB to create ground truth data is fast and accurate and does not require much manual labor. One drawback of using ORB is that it cannot be controlled where the ground truth pixels are calculated. If a key-point matching algorithm can be designed for this specific purpose this technique has potential to simplify development of an accurate stereo camera depth sensor. As mentioned earlier, even though ORB can compute accurate disparities the algorithm is not fast enough, and the data are too sparse to compete with CNNs in a depth sensor application.

4.5.9 Future work

The potential for a stereo camera and computing unit to be used as a depth sensor have been evaluated. With the assumptions that better networks and hardware setups exist or will be created in the future there are good possibilities to further develop the proposed concept into a high-functional system. We propose that a depth sensor unit is created with three cameras and a computing unit with similar performance as TX2 [18]. The cameras should be positioned such that both a

short and long baseline can be obtained leading to a larger distance interval that depth can be predicted in. Then the best available network with respect to depth measurements, speed and memory can be trained self-supervised. Implementing adaptive behavior and vision in dark with IR light are two other possibilities, both with promising performance. Finally, the system should be evaluated with respect to depth measurements and not only disparity measurements to gain information of the true performance. The presented methods using LiDAR and a matching algorithm (ORB) are two possibilities to use for evaluation of the depth sensor.
Areas where the proposed concept have good potential are auto-guided vehicles where depth in combination with tracking can prevent accidents. This combination could also enrich surveillance cameras with a third dimension such that world positions and velocities can be calculated in for example production facilities. Lastly, the suggested concept could be used to create dense 3D scans of both light and dark environments in a fast and detailed way. One possible implementation of this would be to create a full-HD RGB-D scan of a mine or production facility both fast and effective since two stereo images in full-HD as well as a depth map of 2 million depth measurements can be obtained every second or faster.

5
Conclusion

We have introduced a concept of using a stereo camera and computing unit as a depth measurement sensor. This was done by implementing two neural network architectures, SH-Net and XCNN, with a self-supervised loss-function and training on data collected with a specific camera rig. Furthermore, an adaptive behavior has been implemented making it possible for the system to be constantly improving while used as a depth sensor. Night vision through IR light have also been implemented with promising results. With custom created evaluation methods, the depth estimations have been evaluated and it was shown that low disparity errors are not equivalent with low depth errors. Longer baselines are less sensitive to disparity-errors but at the cost of the closest measurable distance. The suggestion is to combine two or more baselines in order to obtain qualitative measurements for a larger interval. Moreover, two evaluation metrics have been proposed that either by using LiDAR measurements or with the use of a feature matching algorithm such as ORB can verify the accuracy. These metrics as well as proposed hardware and software setups can be used to evaluate a real product and guide the development of such a system.
The proposed method can produce 2 million depth measurements from two full-HD color images at 20 FPS with SH-Net and 18 FPS with XCNN on a GeForce RTX3090 with 10,496 CUDA cores. On the smaller Jetson TX2 platform the corresponding frequencies are 0.50 FPS for SH-Net and 0.44 FPS for XCNN. For the optimal combination of a 42 cm baseline and visible distances between 3 and 20 meters the SH-Net reached an absolute mean error of 3.26 meter in daylight on board a moving vehicle. For a static position and with IR as only source of light the SH-Net had a mean error of 0.97 meter for the same distances. Therefore, both usage in static positions in darkness and light as well as on moving vehicles are possible implementations with promising results. The depth measurement accuracy is lower than already existing methods, for example LiDAR, but with the advantage of retrieving much more data per second. We propose that future work develop more lightweight and robust network architectures and combine with the proposed hardware setups. Then a competitive sensor unit can be developed that produce much data in a fast and accurate way. Future work should also focus on finding smart ways to construct dense ground truth data for evaluation. If the accuracy of the depth sensor cannot be verified the applications become limited.

Bibliography

[1] Assisted Docking System and self docking [online] Available at: https://www.volvopenta.com/assisteddocking/ [Accessed 21 may 2021].

[2] Volvo pioneers autonomous, self-driving refuse truck in the urban environment [online] Available at: https://www.volvogroup.com/en/news-and-media/news/2017/may/news-2561936.html [Accessed 21 may 2021].

[3] Optimizing productivity through complete autonomous solutions [online] Available at: https://www.volvoautonomoussolutions.com/en/our-solutions [Accessed 21 may 2021].

[4] Tesla autopilot [online] Available at: https://www.tesla.com/autopilot [Accessed 21 may 2021].

[5] J. Knight, "Safety critical systems: challenges and directions," in *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, pp. 547–550, 2002.

[6] R. Andersson and O. Noresson, "Utilization of quadnocular stereo vision for si-multaneous localization and mapping in autonomous vehicles," Master's thesis, Chalmers tekniska högskola, https://hdl.handle.net/20.500.12380/219226, 2015. An optional note.

[7] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, "End-to-end learning of geometry and context for deep stereo regression," *CoRR*, vol. abs/1703.04309, 2017.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[9] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[10] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," *CoRR*, vol. abs/1803.08669, 2018.

[11] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "Guided aggregation net for end-to-end stereo matching," vol. abs/1904.06587, 2019.

[12] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4040–4048, 2016.

[13] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," *CVPR*, vol. abs/1609.03677, 2017.

[14] Y. Zhong, Y. Dai, and H. Li, "Self-supervised learning for stereo matching with self-improving ability," vol. abs/1709.00930, 2017.

[15] L. Peng, D. Deng, and D. Cai, "Geometry-based occlusion-aware unsupervised stereo matching for autonomous driving," *CVPR*, vol. abs/2010.10700, 2020.

[16] T. Gröndahl and A. Samuelsson, "Self-supervised cross-connected cnns for binocular disparity estimation," Master's thesis, Chalmers University of Technology / Department of Mechanics and Maritime Sciences, 2018.

[17] W. Gan, P. K. Wong, G. Yu, R. Zhao, and C. M. Vong, "Light-weight network for real-time adaptive stereo depth estimation," *Neurocomputing*, vol. 441, pp. 118–127, 2021.

[18] NVIDIA Developer. 2020. Harness AI At The Edge With The Jetson TX2 Developer Kit. [online] Available at: `https://developer.nvidia.com/EMBEDDED/jetson-tx2-developer-kit` [Accessed 23 December 2020].

[19] N. Qian, "Binocular disparity review and the perception of depth," 1997.

[20] M. Santoro, G. AlRegib, and Y. Altunbasak, "Misalignment correction for depth estimation using stereoscopic 3-d cameras," in *2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 19–24, 2012.

[21] H. Hirschmuller, *Accurate and efficient stereo processing by semi-global matching and mutual information*. Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[24] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," *CoRR*, vol. abs/1707.02131, 2017.

[25] H. J. Kelley, "Gradient theory of optimal flight paths," *Ars Journal*, vol. 30, no. 10, pp. 947–954, 1960.

[26] E. J. Kirkland, *Bilinear Interpolation*, pp. 261–263. Boston, MA: Springer US, 2010.

[27] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[28] Wikipedia contributors, "Infrared — Wikipedia, the free encyclopedia," 2021. [Online; accessed 24-May-2021].

[29] J.-E. Källhammer, "Imagingthe road ahead for car night-vision," *Nature Photonics - NAT PHOTONICS*, vol. sample, pp. 12–13, 09 2006.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, pp. 2564–2571, 2011.

[31] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*. Int. J. Comput. Vision, vol. 60, no. 2, 2004.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," vol. 3951, pp. 404–417, 07 2006.

[33] D. Viswanathan, "Features from accelerated segment test (fast)," 2011.

[34] M. Calonder, V. Lepetit, and P. Fua, "Brief: Binary robust independent elementary features," 12 2011.

[35] Li-imx185-mipi-cs, Leopard Imaging. 2021. [online] Available at: https://www.leopardimaging.com/product/csi-2-mipi-modules-i-pex/csi-2-mipi-modules/rolling-shutter-mipi-cameras/2-36mp-imx185/li-imx185-mipi-cs/ [Accessed 30 Mars 2021].

[36] Sf30-c high speed laser rangefinder - 100 m, PARALLAX. 2021. [online] Available at: https://www.parallax.com/product/sf30-c-high-speed-laser-rangefinder-100-m/ [Accessed 30 Mars 2021].

[37] NVIDIA GeForce RTX 3090-grafikkort [online] Available at: https://www.nvidia.com/sv-se/geforce/graphics-cards/30-series/rtx-3090/ [Accessed 30 Mars 2021].

[38] M. Menze, C. Heipke, and A. Geiger, "Joint 3d estimation of vehicles and scene flow," in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[39] MATLAB, *Matlab Stereo Camera Calibrate App (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[40] G. R. Mueller and H. Wuensche, "Continuous extrinsic online calibration for stereo cameras," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 966–971, 2016.

[41] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[42] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[43] R. Andersson and O. Noresson, "Utilization of quadnocular stereo vision for simultaneous localization and mapping in autonomous vehicles," 2015.

A

Appendix 1

The detailed network architecture for SH-Net can be found in Table A.1. Kernel size, number of features, stride and dimensions can be read for each layer in the networks.

	Layer Description	Output Dimensions
	Input Images	W, H, C
	Encoder 1	
1-3	[3 x 3, 5 x 5, 3 x 3] conv, 32 features, stride 1	W, H, 32
4-9	3 x 3 conv, [48, 64, 96, 128, 128, 128] features, stride 2	W/128, H/128, 128
	Decoder 1	
10	4 x 4 deconv, 128 features, stride 2	W/64, H/64, 128
11	3 x 3 conv, 128 features, stride 1	W/64, H/64, 128
12-22	repeat 10-11 with features [128, 128, 96, 64, 48, 32, 32]	W/2, H/2, 32
	Encoder 2	
23	3 x 3 conv, 48 features, stride 2	W/4, H/4, 48
24	3 x 3 conv, 48 features, stride 1	W/4, H/4, 48
25-35	repeat 23-24 with features [64, 96,128, 128, 128]	W/128, H/128, 128
	Decoder 2	
36	4 x 4 deconv, 128 features, stride 2	W/64, H/64, 128
37	3 x 3 conv, 128 features, stride 1	W/64, H/64, 128
38-49	repeat 36-37 with features [128, 128, 96, 64, 48, 32]	W/2, H/2, 80
	Output layer	
50	5 x 5 deconv, 2 features, stride 2	W, H, 2

Table A.1: SH-Net network architecture in detail.

B

Appendix 2

The detailed network architecture for XCNN can be found in Table B.1. Kernel size, number of features, stride and dimensions can be read for each layer in the networks.

	Layer Description	Output Dimensions
	Input Images	W, H, C
	Convolutional layers	
1	7 x 7 conv, 20 features, stride 2	W/2, H/2, 20
2-3	5 x 5 conv, 20 features, stride 1	W/2, H/2, 20
4	5 x 5 conv, 20 features, stride 2	W/4, H/4, 20
5-6	3 x 3 conv, 20 features, stride 1	W/4, H/4, 20
7	3 x 3 conv, 40 features, stride 2	W/8, H/8, 40
8-9	3 x 3 conv, 40 features, stride 1	W/8, H/8, 40
10	3 x 3 conv, 40 features, stride 2	W/16, H/16, 40
11-12	3 x 3 conv, 40 features, stride 1	W/16, H/16, 40
13	3 x 3 conv, 80 features, stride 1	W/32, H/32, 80
14-15	3 x 3 conv, 80 features, stride 1	W/32, H/32, 80
	Deconvolutional Layers	
16	3 x 3 deconv, 80 features, stride 1	W/32, H/32, 80
17	3 x 3 deconv, 80 features, stride 2	W/16, H/16, 80
	residual connection layer 17 and 13	
18	3 x 3 deconv, 40 features, stride 1	W/16, H/16, 40
19	3 x 3 deconv, 40 features, stride 2	W/8, H/8, 40
20	3 x 3 deconv, 40 features, stride 1	W/8, H/8, 40
	residual connection layer 19 and 9	
21	5 x 5 deconv, 20 features, stride 2	W/4, H/4, 20
22	3 x 3 deconv, 20 features, stride 1	W/4, H/4, 20
23	5 x 5 deconv, 20 features, stride 2	W/2, H/2, 20
24	3 x 3 deconv, 20 features, stride 1	W/2, H/2, 20
25	5 x 5 deconv, 20 features, stride 2	W, H, 20
	Output layer	
26	5 x 5 conv, 1 feature, stride 1	W, H, 1

Table B.1: XCNN network architecture in detail.

DEPARTMENT OF MECHANICS AND MARTITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY