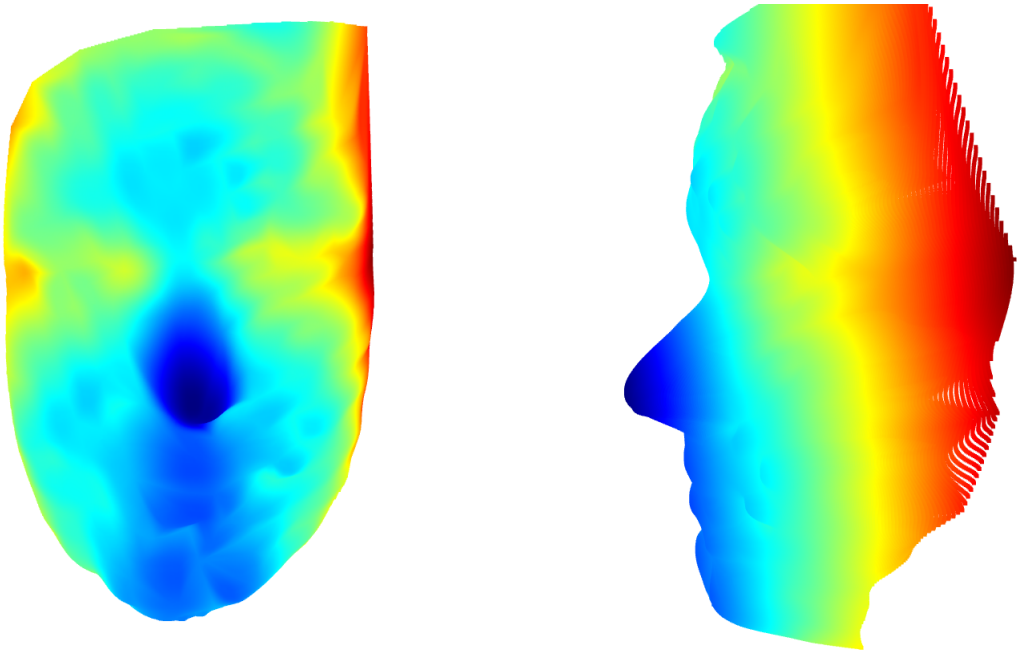
CHALMERS
UNIVERSITY OF TECHNOLOGY



Structured Light Based Depth and Pose Estimation

Improving Structured Light Based Depth and Pose Estimation Using CNNs

Master's thesis in Biomedical Engineering

REBECCA JONASSON ANNA KOLLBERG

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Master's thesis 2019

Structured Light Based Depth and Pose Estimation

Improving Structured Light Based Depth and Pose Estimation Using CNNs

REBECCA JONASSON
ANNA KOLLBERG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical engineering
Chalmers University of Technology
Gothenburg, Sweden 2019

REBECCA JONASSON
ANNA KOLLBERG

Supervisors: Amir Shahroudy, Department of Electrical Engineering, Henrik Lind, Smart Eye AB
Examiner: Fredrik Kahl, Department of Electrical Engineering

Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Depth map of a face using structured light visualized in 3D.

Typeset in LaTeX

Improving Structured Light Based Depth and Pose Estimation
REBECCA JONASSON
ANNA KOLLBERG
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Estimation of head pose and distance to the driver is crucial in the area of active safety. Systems can detect inattention of a driver by studying the head pose, in order to alarm the driver before an accident occurs. Structured light for depth estimation can be implemented in an one camera system with low cost and high accuracy. Furthermore, using convolutional networks for pose and depth estimation is a broad area of research.

The aim of the project was to investigate and implement algorithms for constructing depth maps using structured light. The depth estimates were compared with a current single point depth estimate, regarding both accuracy and invariance to characteristics of the head. The project also aimed to investigate how depth and pose estimation performed by CNNs were impacted by adding depth maps as input.

The resulting algorithm constructed depth maps of faces using structured light. The accuracy of the algorithm for estimating depth using structured light was evaluated to be less than one centimeter and the obtained precision was considered high. Furthermore, the performance of the algorithm was invariant to head characteristics.

The impact of adding depth maps as input to convolutional networks was studied, both using multitask networks and singletask networks. The singletask networks predicted either depth or rotation whilst the multitask networks predicted depth, xy-position, rotation and landmarks. Before training any networks, a data collection was carried out to obtain all required inputs.

Adding depth maps as additional input to a singletask network for prediction of depth or rotation, compared to using only IR images as input, increased the performance. Furthermore, including depth maps as input to a multitask network was found to increase the performance of prediction of landmarks, xy-position and rotation while no significant difference could be concluded for prediction of depth.

Keywords: Depth map, structured light, computer vision, convolutional networks

Acknowledgements

First of all, we would like to thank our examiner Fredrik Kahl and our supervisors at Chalmers and Smart Eye for making this project possible. We would also like to thank for any help we have gotten by employees at Smart Eye with teaching us about current systems, how the networks are implemented etc. Furthermore, thank you Lina Olandersson and Susanna Larsson, students at Chalmers University of Technology, for valuable feedback regarding our report.

Lastly, a big thank you to all who volunteered to be part of our data collection.

Rebecca Jonasson & Anna Kollberg
Gothenburg, June 2019

Contents

List of Figures

List of Tables

1
Introduction

Active safety is a broad area of research, with rapidly increasing importance as the technology of autonomous driving evolves. Simultaneously, the need for a deeper understanding of the driver's behaviour in traffic arises when aiming to implement advanced driver assistance systems to make traffic more safe and less unpredictable. Such driver assistance systems can alert the driver when needed, which serves to reduce the risk of traffic accidents caused by driver inattention. According to the Swedish National Road and Transport Research Institute VTI [1] studies imply that tiredness may be a contributing factor to 10-40% of all traffic accidents, making further research within this area of active safety highly motivated. In order to estimate driver awareness, properties like head pose, eye movements and body motion are of use. Such properties can be used to detect driver fatigue in real time, for instance by identifying pixels corresponding to the eyeball in a consecutive set of frames captured by a camera in the compartment [2]. Furthermore, by evaluating the head pose driver inattention can be detected and the driver can be alarmed if necessary. Yet another aspect where head pose could be of interest is regarding the execution of air bags. A detailed map of the position of the driver could prevent the release of air bags if the driver is in a position where such a release could be harmful.

Since systems for eye tracking and pose estimation for advanced driver assistance systems presented in research have been found to performed well, the industry has started to incorporate such techniques in the development of new products. Smart Eye [3] is a company that produces advanced systems for eye tracking. In addition to track eye gaze and eye movements, the devices developed by Smart Eye can also estimate the pose of the head as well as the distance to the head. Smart Eye has systems consisting of one single camera or a set of multiple cameras. For automotive applications the single camera system is considered well suited as it is smaller in size and easier to incorporate into the compartment.

1.1 Project description

Today, camera devices developed by Smart Eye have technology to track depth, xy-position, rotation and facial landmarks of a driver in a compartment using a single camera. Though, the tracked data is estimated based on the characteristics of the head of the driver, making it sensitive to deviations in size and shape. As a result, the single point depth estimate has an accuracy of 10% of the distance, which is not considered accurate enough. Moreover, in order to achieve this accuracy it is necessary to include small head movements as an initial step of the recording,

such that the head model which is used to create data from tracking can be built properly. The goal is to increase the accuracy of the single point depth estimate to less than one centimeter. Another goal is to create depth maps of the face, since the current depth estimate only consists of one single point. The single point estimate is positioned between the eyes, behind the nose bridge, whilst structured light will be implemented to model the distance to the surface of the face. Thereby, an offset between these two measures corresponding to the distance between them needs to be considered in order to enable a comparison.

Currently, infrared images, IR image, are used as input to convolutional neural networks, CNNs, for various task, including estimation of depth and the head pose of the driver. The networks are trained supervised using the output from the tracking algorithms as ground truth. As structured light will be implemented to create depth maps of the face, the possibility to increase the performance by adding depth maps as additional input will be explored. The ground truth parameter for depth will be replaced by the depth value computed using structured light, whilst ground truth for xy-position, rotation and facial landmarks is computed from the current tracking system.

Furthermore, the tracking algorithms are based on IR images. Hence, the output of the tracking algorithms can not be generated from images contaminated by IR structured light. Therefore, it must be possible to obtain regular IR images in every other frame of the camera in order to obtain the required output used as ground truth to the network. As a result, a prototype camera called Smart AI-X 3D has been designed.

1.2 Aim

The aim of the project was to investigate and implement algorithms for constructing depth maps using structured light. The depth estimates were compared with a current single point depth estimate, regarding both accuracy and invariance to characteristics of the head. The project also aimed to investigate how depth and pose estimation performed by CNNs were impacted by adding depth maps as input.

1.3 Specification of issue under investigation

The project is divided into two parts: to compute depth maps using structured light and to improve estimation of depth and pose by adding depth maps as input to neural networks. The following questions will be answered as the project proceeds:

1.3.1 Part 1: Generation of depth maps using structured light

- Is it possible to obtain an accuracy for the depth estimate of less than one centimeter when using structured light?
- Is it possible to obtain an accuracy and precision in computed depth which is independent of the characteristics of the head using structured light?

1.3.2 Part 2: Depth and pose prediction using depth maps

- Can the performance of the CNNs performing pose and depth estimation improve by using depth maps computed from structured light as additional input?

1.4 Limitations

The algorithms developed during the project are limited to be based on a specific camera developed by Smart Eye during the project, namely the Smart AI-X 3D. Until the camera is available, initial testing will be performed on another camera, with different pattern and software. This report is limited to discuss results obtained from the Smart AI-X 3D, wherefore initial testing on any other camera will be exluded from the report.

The project is limited to use only one near IR camera and one projector. The projector used for depth estimation is limited to use a uniform grid of dots as pattern. The accuracy of the generated depth estimations is limited to be compared to present technologies used for depth estimation at Smart Eye. Furthermore, the evaluation of the accuracy is limited to use available measurement tools, which are safe for humans. No depth maps are available for comparison with the ones obtained from the implemented depth estimation algorithm.

There is a limitation related to the difference in origin of the two depth estimations. The offset between these two depth values is limited to be 1 cm for all depth estimations, in both the comparison of the accuracy and when updating the ground truth in the CNNs.

The second part of the project is limited to improve depth and head pose prediction using the obtained depth information from the first part of the project. Data for the CNNs will be limited to recordings of employees at Smart Eye. All work concerning the networks is limited to be based on the current code framework at Smart Eye.

The evaluation of Part 1 as well as the data collection for Part 2 is limited to office environment, and no other lighting conditions are tested. The test persons in the data collection are not wearing glasses or similar accessories. The evaluation of the depth estimate of Part 1 is limited to be tested on two different persons, at distances between 50 and 75 cm. The images used for evaluation are taken in front view.

Another limitation is that both the frame used for obtaining output from current tracking algorithms and the consecutive frame with projected structured light are captured using an IR camera. Hence, the frames need to be taken separately, which results in a time difference on $1/fps = 1/60s$ between the frame used for generating the depth information and the frame used for obtaining ground truth.

The area of active safety will be considered and a brief discussion of social and ethical aspects of possible implementations will be carried out.

1.5 Contributions

There are two main contributions of this project. First, the project aims to create depth maps based on the projection of a uniform IR pattern, instead of a pseudo random pattern which is frequently implemented for this task. The second main contribution is to study the impact of adding depth maps of the face as input to a convolutional network for prediction of depth and head pose, as compared to only using IR images as input.

2
Theory

In the following sections a literature review of theories and studies relevant to this project are presented.

2.1 Depth estimation

In order to obtain a correct 3D representation of a scene, regular 2D images are not enough due to lack of depth. Instead, 3D surface imaging must be deployed to gather all data necessary for such a representation. This has been subject to numerous research papers during the years, especially as products implementing the techniques have become available at low cost. The process consists of approximating a depth map of a scene, which subsequently can be used for 3D reconstruction of objects of interest. Several computer vision techniques can be applied to reach a depth estimate and the corresponding camera system must be designed accordingly. The three most common techniques for obtaining depth maps are time of flight, stereo vision and structured light. A brief description of time of flight and stereo vision is given in the following section. Then, the concept of structured light is described followed by a comparison of advantages and disadvantages of structured light as compared to the two other techniques described. Lastly, learning techniques for estimation of depth maps are presented.

2.1.1 Time of flight

Time of flight cameras, ToF cameras, utilize the speed of electromagnetic radiation in air [4]. It is trivial that the distance $D[m]$ can be found from the formula $D = c\tau$ where $c \approx 3 \cdot 10^8[m/s]$ is the speed of radiation and $\tau[s]$ the time. Now, let radiation be emitted at time 0 by a ToF transmittor and reflected back on the surface. The radiation will then reach the ToF receiver at time τ and the distance it has traveled is $2D$. The distance can be found from:

$$D = \frac{c\tau}{2}$$

An advantage of ToF systems, compared to structured light and stereo vision systems, is that ToF suffers less from occlusion [4]. The reason for this is that the transmitter and receiver are designed such that they are collinear, i.e. placed on a line. This is achieved by placing the transmitter close to the receiver.

A problem of the ToF system is how to measure the time. As an example, in order to cover a distance of one meter a precision of nanoseconds is needed, whilst for

centimeter precision picoseconds is needed. One method for handling this problem is continuous wave modulation, described below.

Microsoft has developed range sensing devices called Kinect™. In 2010 Microsoft released Kinect™ based on structured light for range sensing. A few years later, Microsoft created Kinect One based on ToF. In order to use time of flight, continuous wave intensity modulation is used in the device. This is also what is most often used in ToF systems. The scene is illuminated using a near infrared intensity modulated periodic light which causes a time shift in the optical signal. The time shift in the optical signal corresponds to a phase shift in the periodic signal, which is detected in each sensor pixel. Finally, the time shift is transformed into a distance.

2.1.2 Stereo vision

The basic stereo vision system consists of two cameras, a camera to the left called reference camera and one to the right called target camera [4]. The two cameras need to be calibrated and put into a 3D and 2D reference system. The 2D reference system has coordinates (u_L, v_L) for the left camera and (u_R, v_R) for the right. The disparity, d, can be found from:

$$d = u_L - u_R$$

i.e. the disparity is the difference in horizontal coordinates in the 2D reference system.

From the disparity the depth in the 3D reference system for each pixel can be computed [4]. The formula for computing depth is:

$$D = \frac{f \cdot b}{d \cdot p} \tag{2.1}$$

where f is the focal length [m], b represents the baseline [m] i.e. the distance between the cameras, d is the disparity value described above, p the pixel size [m] and D is the depth [m]. In Figure 2.1 the idea of stereo vision is illustrated.

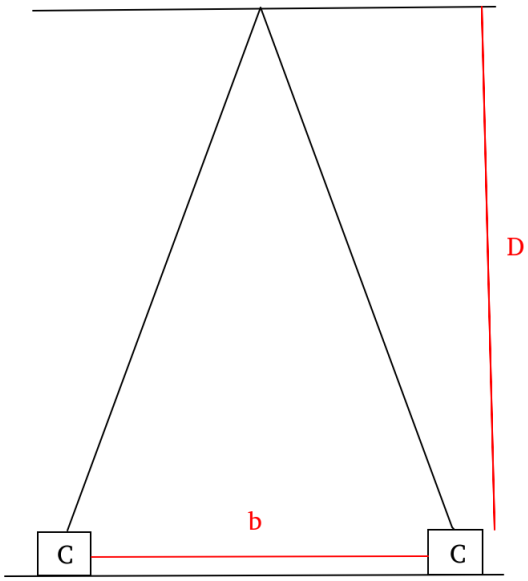


Figure 2.1: Overview of stereo vision where b is baseline, D is depth and c is camera.

Stereo vision is not a new technique within computer vision, already in 1981 an image registration technique was proposed which could be used in a stereo vision system [5]. Moreover, in 1998, stereo vision was used by Bertozzi et al. [6] to perform generic obstacle and lane detection system on moving vehicles. For obstacle detection Bertozzi et al. used stereo images, in order to directly detect the presence of obstacles.

Stereo vision for urban 3D semantic modelling was studied by Sengupta et al. [7] in 2013. Their proposed algorithm generated a 3D reconstruction with semantic labellings, where the input was street level stereo image pairs from a camera on a vehicle in motion. Stereovision was used to generate depth maps, which in turn was fused into global 3D volumes. The labelling was performed using a framework called Conditional Random Field and the label estimates were aggregated to annotate a 3D volume.

2.1.3 Analytic depth resolution of stereo vision

In order to further investigate the resolution of the depth estimation using Equation 2.1 the spatial sampling of the image must be considered [4, 8]. This is because the depth estimation is highly dependent on the estimation of distances in the image. Following the derivation in [8], an error δd is introduced in the estimated position of a certain point imaged by the camera and the measurement of Δd can be expressed as:

$$\Delta d = \Delta d_0 \pm \delta d$$

Using Δd_0 as the true disparity of the point, the distance to a certain point can be described as:

$$D = \frac{bf}{\Delta d_0}$$

Equivalently, Δd_0 can be expressed by:

$$\Delta d_0 = \frac{bf}{D} \tag{2.2}$$

Though, as an error in position estimation in turn introduces an error δD in depth, the depth measurement can be expressed as $D = D_0 \pm \delta D$. Again, D_0 is assumed to be the true depth. Now, Taylor expansion can be used to express δD in terms of δd:

$$\delta D = \frac{bf}{\Delta d_0^2}\delta d$$

Finally, substituting Δd_0 by the expression in Equation 2.2 yields the following depth resolution model:

$$\delta D = \frac{D^2}{bf}\delta d \tag{2.3}$$

where Δd is the resolution of disparity.

As the disparity resolution is most easily measured in pixels, the focal length needs to be converted to pixels as well. Hence, focal length is defined to be $f = f_m/p$ where f_m is the focal length in unit meter and p is the pixel size in unit meter.

2.1.4 Structured light

Structured light is an optical method to measure objects, consisting of a camera system in combination with a projector, as illustrated in Figure 2.2. The process of a structured light system mimics stereo vision but here disparity is estimated between two projected patterns instead of two distinct cameras [4]. The purpose of the projector is to emit an encoded pattern onto a scene, which then is captured by the camera. Since the emitted pattern gets altered while projected onto objects the displacement of the pattern from its original counterpart can be used to obtain a disparity map. The disparity is then used to estimate depth in the scene equally to stereo vision. The camera system usually consist of one or two cameras. Two cameras often increase the performance of the system as many problems of structured light systems can be dealt with more efficiently. For instance, two cameras make the system less sensitive to non-idealities in sensors and projectors but also less sensitive to ambient light, as the two patterns compared will be affected by these factors to the same extent.

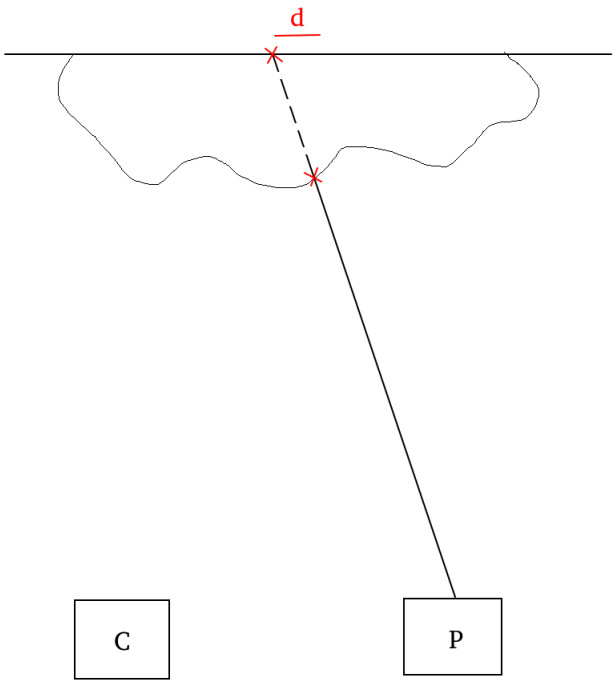


Figure 2.2: Overview of structured light where p is projector, c is camera and d is disparity.

As the light source projects a pattern onto the surface of the object, two crucial steps in the process of using a structured light system consist of encoding and decoding of the projected pattern [9]. Considering the characteristics of the pattern it should be possible to find correspondences between a pattern and its projected counterpart. A wide set of patterns can be used for structured light systems, mostly different versions of stripe patterns or grid patterns. In order to make it easier to distinguish between different parts of a pattern distinctions in intensities, phases and sizes can be imposed. Depending on the characteristics of the encoded pattern different decoding algorithms can be applied, such as pattern recognition for pseudo-random grids.

To obtain a depth estimate from the disparity between patterns the formula in Equation 2.1 should be modified according to the calibration of the structured light system [4], which yields:

$$\Delta D_i = \frac{1}{\frac{1}{D_\mathrm{c}} + \frac{d_{\mathrm{rel}_i}}{bf}} - D_\mathrm{c} \tag{2.4}$$

In the formula above, a difference in depth ΔD_i is estimated using the depth of the calibration image D_c and the disparity between points at the distance D_i and the points at the calibration distance, called d_{rel_i}. Then, the depth at a pixel i is obtained by:

$$D_i = D_\mathrm{c} + \Delta D_i$$

The depth resolution for structured light is computed using the same formula as for stereo vision, Equation 2.3. Trivially, this can be proved following the steps in Section 2.1.3.

Structured light techniques are frequently used to obtain depth maps. As mentioned in previous sections, structured light was implemented in the first version of Microsoft Kinect [10] where depth maps are used for gesture recognition and human pose estimation.

Moreover, since structured light can be used to obtain measurements of an object without contact with its surface the system has been considered well suited for a wide set of reconstruction applications, including forensic medicine [11] and reconstruction of cultural relics [12].

2.1.5 Advantages and disadvantages of structured light

As there exist several techniques to estimate 3D surfaces it is of interest to compare the advantages and disadvantages of structured light techniques compared to other techniques, mainly time of flight and stereo vision as these are the most common.

The main advantage of structured light is that the system can be designed with only one camera as compared to stereo vision, which can drastically reduce the production cost. It also makes the system more portable and easier to incorporate. An additional advantage of structured light is that it can be used to accomplish high spatial resolution, since it does not require processing at sensor level like in the case of ToF.

Since the projector and the IR camera are placed at distinct locations on the device, some of the projected dots may not be seen by the camera due to occlusion. The problem of occlusion arises both in structured light and stereo vision [10]. As for structured light, this can result in problems in the decoding process of the pattern since data is lost. In order to find corresponding patterns it is therefore useful to identify which part of the pattern that have been occluded from the projector and neglect the corresponding parts of the original pattern. Moreover, occlusion results in holes which often require further post processing. Since ToF cameras only have one single viewpoint this technique does not suffer from problems related to occlusion.

Another parameter that affects the performance of structured light is the presence of ambient illumination since this can result in a corrupted pattern. The problem can occur even when using infrared light, since the wavelength of the projector can coincide with the spectrum of wavelengths for sunlight. Therefore, it might not be possible to filter from the signal [13], making structured light nonrobust to ambient illumination. Contrary, this problem does not appear when using time of flight cameras which makes it more suitable for outdoor measurements than structured light. Though, both techniques suffer when the environment is too bright since this can cause over-saturation, which also is the case for stereo light.

All systems using cameras can be affected by noise and sensor non-idealities. The presence of such phenomena may be extra significant for single camera structured light systems as these factors highly affect the difference between compared patterns.

Finally, due to absorption and reflectivity properties of objects in the scene the projection of the pattern may suffer from severe color and intensity distortions, leading to a major decrease in correctly matched parts of the pattern [4]. Clearly, this is a more frequently appearing issue when encoded patterns that are not uniform

in color or intensity.

By the advantages and disadvantaged stated above it is clear that when choosing a depth imaging technique the production cost must be set in relation to possible reductions in performance caused by technique specific challenges.

2.1.6 Learning techniques for estimation of depth maps

Another possible approach to obtain an estimation of depth is to incorporate learning techniques in combination with monocular images. Depth estimation based on only one image is a challenging task, since it is only possible to obtain local features in the image without any information about its global scale. Features which are possible to extract from monocular images are for instance changes in texture, shading and occlusion but based on the characteristics of the image these may not be enough. As a consequence, information from a single image often result in an insufficient depth estimation. Though, estimating depth using monocular images is desirable since it could reduce the cost of hardware for devices designed to estimate depth and possibly simplify the processes of these. To circumvent these challenges, and since standard techniques such as triangulation can not be applied, neural networks has been presented as a possible solution.

A study within this area of research was performed by Saxena et al. [14] at Stanford University. They presented a model based on supervised learning to predict a depth map with only monocular images as input and used a hierarchical multiscale Markov Random Field, MRF. Such a random field consist of random variables which all have Markov properties, i.e. their future state depends only on the present state and no other previous states. In the model presented by Saxena et al., the MRF incorporated both local and global features of the image to estimate depths as well as to estimate relations between depths at distinct points of the image. Hence, both absolute depths features and relative depths features were included. The proposed model was able to estimate depth for a varying set of scenes. Furthermore, an algorithm which combined depth estimates using triangulation and monocular depth estimates was proposed, which was found to perform better than when only using either of the two depth estimates.

2.2 Pose estimation

The 3D modeling techniques presented in previous sections can be useful for a number of tasks, for instance pose estimation which is the subject of the second part of this project. Head pose estimation in computer vision is the process of inferring the orientation of the human head from an image [15]. Ideally, the estimation shall be invariant to camera distortion, projective geometry, biological appearance and accessories, e.g. glasses and hats. The definition of head pose estimation is wide, it can be anything from a system that identifies the head in frontal versus left/right profile to continuous angular measurements. One option to describe the rotation of the head is by using quaternions, $\mathbf{q}$=(w,x,y,z) [16]:

$$\hat{\mathbf{q}} = (cos\left(\frac{\theta}{2}\right), sin\left(\frac{\theta}{2}\right)\hat{\mathbf{n}})$$

In the equation above, $|\hat{\mathbf{q}}| = 1$. The angle of rotation is represented by θ and the axis of rotation is represented by $\hat{\mathbf{n}} = (n_x, n_y, n_z)$, which has the property $|\hat{\mathbf{n}}| = 1$.

Furthermore, gaze estimation is closely linked with head pose estimation and to accurately predict gaze direction head pose is needed. In order to perform the head pose estimation many different methods has been studied and used throughout the years, these methods varies from geometric methods which uses the location of features to determine pose to convolutional networks that can map an image to head pose estimation. The focus in this section will be on the advances of using convolutional networks for pose estimation.

2.2.1 Convolutional networks for computer vision

The most commonly used networks for computer visions systems are Convolutional Neural Networks, CNNs. CNNs generally consist of three main neural layers: convolutional layers, pooling layers and fully connected layers. A simple illustration of a convolutional network is visualized in Figure 2.3.

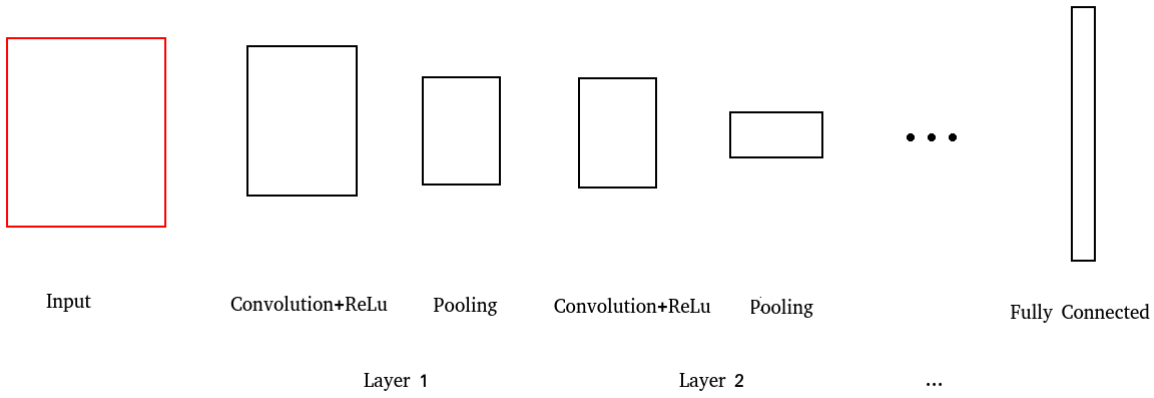


Figure 2.3: Illustration of the typical structure of a convolutional network.

In the convolutional layer, the input image is convolved using filters [17]. The output of this layer is a 2D feature map. The convolutional layer learns correlations between neighbouring pixels and is also invariant to the location of the object. It reduces the number of parameters, wherefore the convolution layer can replace the last fully connected layer for faster learning. The next layer is usually a pooling layer. This layer reduces the dimensions of feature maps and network parameters. Max pooling and average pooling are the two most common pooling layers. The convolutional layers and pooling layers are often repeated throughout the network, ending with fully connected layers. The fully connected layers converts the 2D feature maps into one 1D feature vector.

The training of a neural network consists of two stages [17]. The first stage is called the forward stage where weights and bias for each layer are computed. Using the prediction output and the ground truth a loss is computed. One of the most common loss functions is Mean Average Error, MAE [18]:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\text{error}|_i.$$

In the formula above, the error is computed as a difference between predicted parameters and the ground truth.

In the backward stage, gradients for each parameter are computed and the parameters are updated based on these gradients. The updating of weights depends on the learning rate. A common way of setting the learning rates is using Adam [19], which is an adaptive learning rate method computing individual learning rates. The forward and backward stages are iterated until a stopping criteria is fulfilled. A stopping criteria could be that the loss cost is lower than a certain value or that the number of maximum iterations/epochs, are reached.

A problem of deep learning and large CNNs is overfitting [17]. There are multiple ways to handle this problem, one approach is to use data augmentation, where data is augmented to generate more data.

To evaluate different techniques an annual challenge is held within visual recognition, called ImageNet Large Scale Visual Recognition Challenge, or ILSVRC [17]. In 2012 the CNN AlexNet [20] won this competition. After this breakthrough an increasing number of contestants used deep learning techniques and in 2014 most participants used CNNs as a basis for their models. Between 2014 and 2013 the error in image classification was halved because of this. AlexNet consists of five convolutional layers and three fully connected layers. It utilizes different forms of data augmentation: image translations, horizontal reflections and altering intensities of RGB channels in images.

Other popular CNNs are residual networks, called ResNets [21]. ResNets consist of stacked residual units, with the form:

$$y_l = h(x_l) + F(x_l, W_l)$$

$$x_{l+1} = f(y_l)$$

Here x_l is the input of the l^{th} unit, x_{l+1} is the output of the l^{th} unit, F is a residual function, f is a ReLU function, and $h(x_l)$ is an identity mapping. The residual function F should be learned with respect to $h(x_l)$ by using an identity skip connection. ReLU, Rectified Linear Unit, is an activation function which is both effective and simple and used in many CNNs [22]. A simple illustration of a residual network is visualized in Figure 2.4.

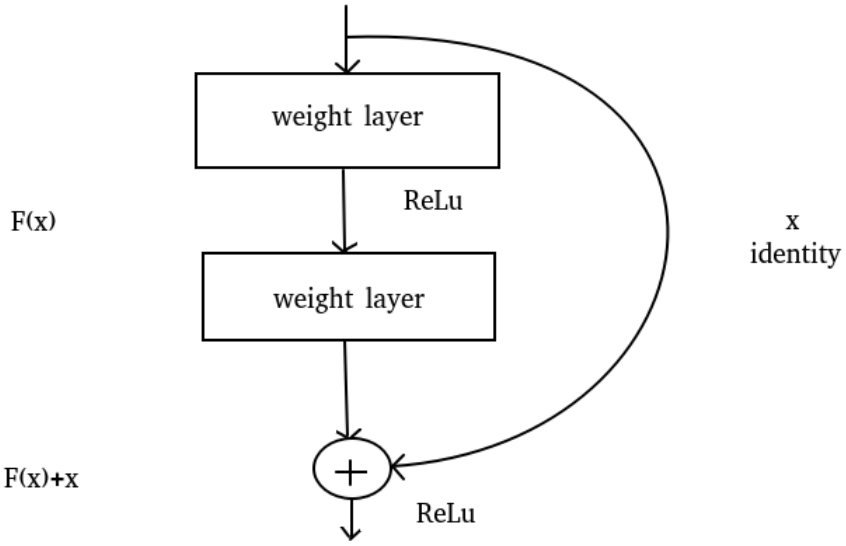


Figure 2.4: Illustration of the typical structure of a residual network.

Generative Adversarial Networks, GANs, is another commonly used network [23]. GANs learn two networks with competing losses. The two networks are called generator and discriminator respectively. The generator network's task is to map a random vector to fake images, and the discriminator's is to distinguish the fake images from the real images. Hence, GANs are frequently used to generate simulated images. The typical structure of GANs is illustrated in Figure 2.5.

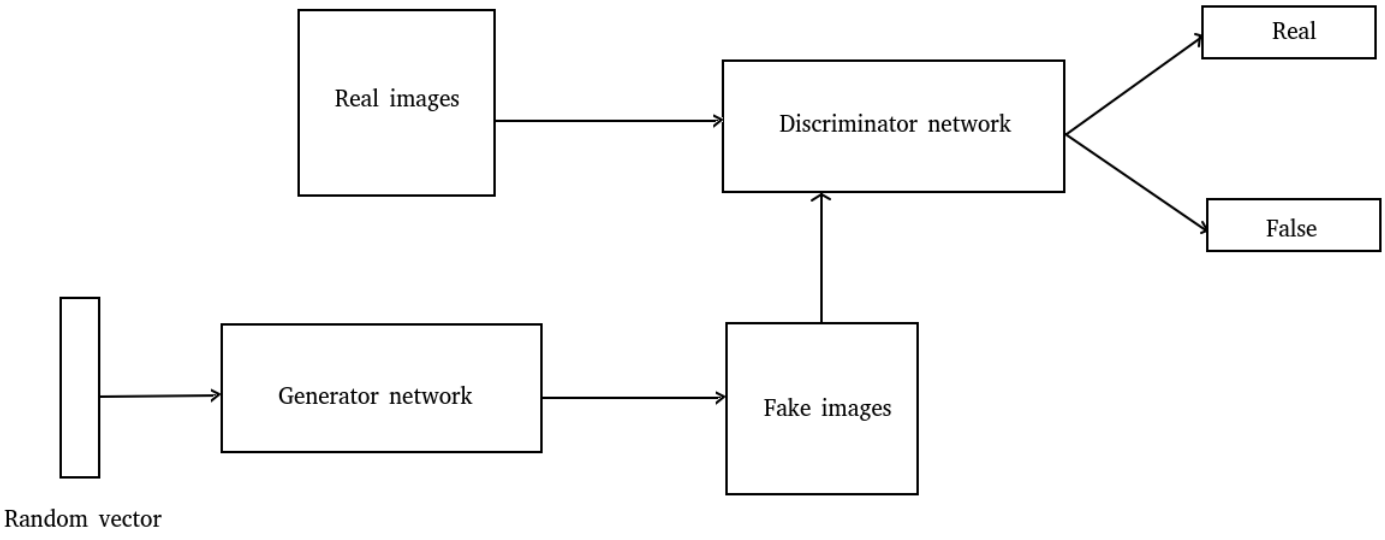


Figure 2.5: Illustration of the typical structure of a generative adversarial network.

Recently yet another network architecture called Facial Attributes-Net [24], or FAb-Net, was presented by the Visual Geometry Group at University of Oxford. The network has an encoder-decoder structure and is a smaller network than ResNet. In the encoder all convolutional layers have size $4\times$ 4 while convolutional layers in the

decoder have size 3×3. The network was implemented to learn facial attributes in a self-supervised approach and performed superior or comparable to state-of-the-art frameworks for similar tasks implementing self-supervised learning.

2.2.2 Depth maps for learning tasks within computer vision

As the techniques to estimate depth have been improved, the number of tasks within the area of computer vision for which depth maps have been found useful has increased as well. For instance, using depth information to perform pose estimation has been found successful. Shotton et al. [25] released a paper in 2013 where they used single depth images for pose recognition in real-time. The idea was to transform the pose estimation problem to a per-pixel classification problem. The input to their system was a 2D representation of a depth image and the output was a 3D skeleton of the person in the image, where 31 body parts were considered. The general objective was to find a function F depending on image I and frame t, where $F(I_t, I_{t-1}, ..) = \theta_t$ and θ_t is the 3D skeleton at frame t. First, for each pixel in the image probabilities for each body part were estimated using randomized decision forests. This was then used to form a body parts image noted as C_t, where each pixel contained a vector with 31 probabilities for each body part considered. Then, a set of joint hypotheses consisting of body part, 3D position and confidence were formed using the probabilities in C_t. Finally, a final 3D skeleton θ_t was estimated by finding combinations of the joint hypotheses which made the skeleton kinematically consistent with the skeleton of the previous frame. With this structure, it was only the very last step that used information from previous frames.

The evaluation was performed on both synthetically generated images and real data. It could be noted that more training data improved the result, up until around 100 000 images. On both synthetic and real data the accuracy was consider high compared to other systems at that time.

Depth information has also been used for human action recognition by Wang et al. [26] in 2016. A deep convolutional neural network with three channels was used together with weighted hierarchical depth motion maps. The method transformed the action recognition problem to an image classification problem, which used information from consecutive depth maps. To make the CNN view-tolerant, the captured depth maps were rotated. Furthermore, different temporal scales were constructed to create a set of distinct spatiotemporal motion patterns. Both of these implementations increased the number of training data. The motion maps were also converted into pseudo color images to enhance the 2D spatial structures. After rotation, the depth maps were projected onto three orthogonal Cartesian planes. Then, for each of the three projected views the absolute differences between subsequent frames were accumulated. Each of the three channels was trained on the projected depth maps independently and the final classification was obtained after fusing all three networks, as a late fusion network. The CNNs were all initialized with models from ImageNet. The method was successful compared to the other methods.

Furthermore, depth maps have been incorporated as input for object recognition tasks. A model for object recognition using both RGB images and depth images has been studied by Eitel et al. [27]. Two separate CNN processing streams were

used for the two inputs, which then were combined using a late fusion network. For the depth data two parts were introduced for more effective results. First, to be able to learn without large depth datasets an encoding of the depth information was used in order to enable usage of pre-trained networks. Here, the depth image was encoded as an RGB image which resulted in information spread to all three RGB channels. A standard pre-trained CNN was then used for recognition. Second, data augmentation was performed for robust learning. In addition, since occlusion often is a problem when working with depth maps, the images were augmented by adding occlusions known from sampling real-world environments. The networks were trained separately for depth and color information as a first stage, both using weights initialized from the ImageNet dataset. In the next training stage the two streams were jointly fine tuned. Finally, classification was performed by a fusion network.

2.2.3 Multitask network for face recognition tasks

There are multiple tasks regarding face recognition, and these tasks can often be performed with higher accuracy if performed simultaneously. Ranjan et al. [28] has implemented and evaluated a multitask learning network for performing face detection, landmark localization, pose estimation and gender recognition simultaneously. Their proposed method was called HyperFace and it fused the intermediate layers of a deep CNN for better performance. HyperFace used three modules. Module one generated class-independent region proposals from the image, module two was a CNN which classified the regions as face or no face and also provided facial landmarks location, gender information and estimated head pose. Module three performed post processing.

Ranjan et al. [28] proposed two different architectures for their network. HyperFace based on AlexNet was their first approach. The network was initialized with the weights of a network called R-CNN-face network, described below. The fully convolutional layers were not needed for pose estimation and landmarks extraction, and was therefore removed. The lower layer features were suitable for pose estimation and landmark detection, whilst the higher layer features were more suitable for the more complex tasks detection and classification. Since their objective was to learn face detection, landmarks, pose and gender simultaneously, they fused features from intermediate layers of the network and learned multiple tasks on top of it. To learn the weights of the network, specific loss functions were used for each task.

To evaluate HyperFace, Ranjan et al. [28] also studied simple CNNs where each CNN performed one task each, e.g. a R-CNN-face network which only performed face detection. Another comparison was made with a model similar to HyperFace, Multitask_Face, which simultaneously detected face, localized landmarks, estimated pose and predicted gender but without fusion of intermediate layers. Instead, Multitask_Face used a fully connected layer at the end of the network to combine the tasks.

After evaluating HyperFace based on AlexNet, Ranjan et al. [28] created a model based on ResNet called HyperFace_ResNet. Geometrical features were again fused from the lower layers and semantically strong features from the deeper layers. Aver-

age pooling was used to generate a shared feature vector. Their evaluation showed that HyperFace and Multi-task_Face were comparable in performance for face detection datasets, which indicated that fusion was not important for the face detection task. For landmark localization, the fusing of layers was more important as HyperFace outperformed Multitask_Face. HyperFace-ResNet also outperformed HyperFace. In pose estimation, HyperFace and HyperFace_ResNet performed better than Multitask_Face. The performance for roll, pitch and yaw differed and yaw was the hardest to estimate. For gender detection there were no distinct differences in performance between MultiTask_Face and HyperFace.

Their conclusion was that all face related tasks are benefited from using a multitask learning framework, and using fusing of intermediate layers improves the performance for pose estimation and landmark localization. HyperFace_ResNet was slower than HyperFace, since it performed more convolutions, but had better performance.

2.2.4 Generating simulated data for pose estimation

When training neural networks data collection is often a problem, as neural networks acquire much data. Shrivastava et al. [23] used synthetic images for training, with both simulated and unsupervised learning to improve the realism of synthetic images. The learning was meant to improve the realism of the simulator's output, whilst preserving the annotation information from the simulator.

The proposed method, called SimGAN [23], used an adversarial network to generate these synthetic images. The network was similar to GAN but instead of using random vectors as inputs synthetic images were used. The first step of SimGAN was to generate synthetic images with a simulator. A network, using ResNet blocks, was used to refine the images from the simulator using a self regularization term which penalized changes between the synthetic and refined image. After training, the refined images should be indistinguishable from real images using a discriminative network.

Pose estimation using SimGAN was evaluated [23]. The training was done on real, synthetic and refined synthetic images and then evaluated on real images. Training on refined synthetic data from SimGAN outperformed both the model trained on real images and the model training on original synthetic data.

3
Methodology

As the project is divided in two parts, the methodology of the project will follow the same structure. First, the specification of equipment is presented in Section 3.1. Secondly, Section 3.2 describes the Part 1 of the project, how to use structured light to form depth maps. Finally, the usage of depth maps as input to CNNs for improved pose estimation is described in Section 3.3.

3.1 Camera and projector specifications

The camera device used for the study was a Smart AI-X 3D camera developed by Smart Eye. The design of the device is shown in Figure 3.1.

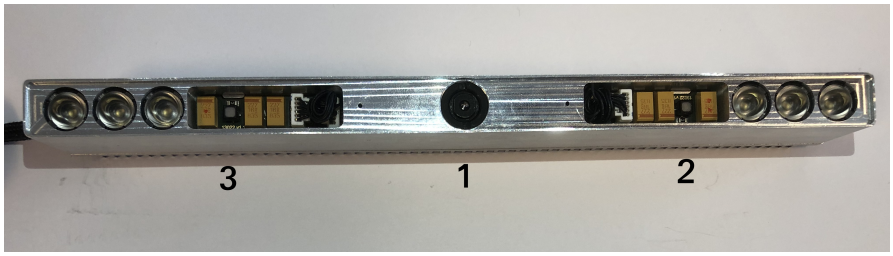


Figure 3.1: Smart AI-X 3D used for this project.

The camera is located in the middle of the device, at location 1. Two BELICE-940 illuminators are incorporated, one on each side of the IR-camera. The projector emits a high contrast uniform and rotated dot pattern of wavelength 940 nm using VCSEL power. There are two types of the BELICE-940, namely type A and type B. Type A emits a 5° rotated pattern while type B emits a 15° rotated pattern, each pattern on average consisting of 4600 dots. The projector used for all images in this project is type A, which is located at position 2 in Figure 3.1 while projector B is located at position 3. As the project is limited to use one projector only, a plastic cover is put on projector B when taking images to exclude its projected pattern at all times. Furthermore, the pattern is projected every other frame, resulting in a generation of both regular IR images and a generation of images of the projected pattern, further described in Section 1.1. The frame rate of the camera is 60 Hz. Camera specifications of the Smart AI-X 3D are listed in Table 3.1.

Camera	Smart AI-X 3D
Focal length [mm]	5.87
Pixel size [μm]	3

Table 3.1: Camera specifications for the Smart AI-X 3D.

3.2 Generation of depth maps from structured light

In the following sections the procedure of obtaining a depth map using structured light is explained in detail. The process was divided into three different parts; preprocessing of images, detection of centroids of dots and estimation of depth. As each part is necessary to achieve the final depth map, all parts must be completed in the correct order and with high enough accuracy to ensure the presence of such a depth map after the final matching of corresponding dots. The inputs to the system was a pair of IR images taken with the Smart AI-X 3D camera. An example of inputs are presented in Figure 3.2 and Figure 3.3. The image shown in Figure 3.2 is the IR image used in the original system to estimate depth, described in Section 1.1, and the projected image shown in Figure 3.3 is what is used to compute the depth map.

Figure 3.2: IR image obtained from the Smart AI-X 3D camera, which is used as input to the tracking algorithms.

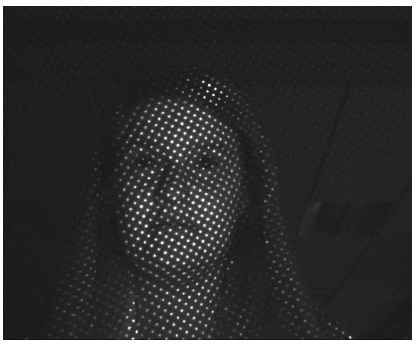


Figure 3.3: IR image with the projected pattern, corresponding to the IR image in figure 3.2. The brightness of the image has been increased.

The calibration image, which can be seen in Figure 3.4, consists of an image of the pattern projected on a flat surface from a distance of 76 cm. This procedure was executed once only, and does not have to be redone as long as neither the projector nor the camera of the device is changed. Since the calibration image has relatively low intensity, an equivalent image with increased brightness is shown in Figure 3.5.

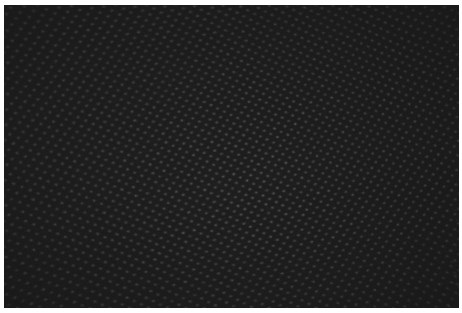


Figure 3.4: Calibration image used in the process of estimating depth.

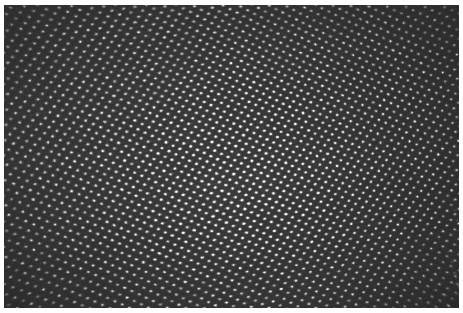


Figure 3.5: Calibration image in Figure 3.4 with enhanced brightness.

3.2.1 Preprocessing of images

All images obtained from the Smart AI-X 3D camera were preprocessed in order to reduce the level of noise in the images. In order to do so, a Gaussian filter was applied as it is a low pass filter. The reduction of noise served to increase the performance of the edge detector implemented to detect dots in the image of the projected pattern, which is further described in the following sections. If the Gaussian filter was not applied the noise level of the obtained depth estimate became significantly higher as compared to when the filter was applied, which is presented in Figure 4.26 to Figure 4.28 in Section 4.2.2.

All frames in the data set were cropped to visualize the head only and omit all other details of the image. To crop the image accordingly, feature points of the face such as chin and cheek had to be found to determine which pixels to omit. When implementing the depth estimation in the framework for Part 2, code for face cropping already implemented at Smart Eye was used. However, when evaluating the depth estimation in Part 1, the face cropping algorithm was not available. Therefore, the images used for the evaluation of depth were simply cropped manually to only visualize the head. The procedure of cropping was not employed for the calibration image, since all information of the calibration pattern must be kept.

3.2.2 Detection of centroids of dots

Finding the centre of each dot in the projected pattern is a crucial step in order to acquire an accurate depth estimate from the provided set of images. A close up of one projected dot of the pattern is shown in Figure 3.6.

Figure 3.6: Close up on one dot in the projected pattern.

As the contrast of the projected dot is relatively low, finding the exact centre of such dot is a challenging task. The process of detecting centroids can be designed in various ways, depending on the characteristics of the projected pattern. As the pattern of interest was uniform, algorithms such as pattern recognition could not be applied. Instead, the centroids of dots were found using edge detection techniques. First, thresholding was applied to each image after preprocessing. Since applied after the noise reduction the thresholding was assured to not increase the level of noise in the image. With the aim of producing an image of only the projected dots while neglecting all other intensities, the threshold value had to be chosen with regard to the intensity of relevant dots of the image. For the calibration image a fix threshold could be found manually and the obtained calibration image after thresholding is shown in Figure 3.7.

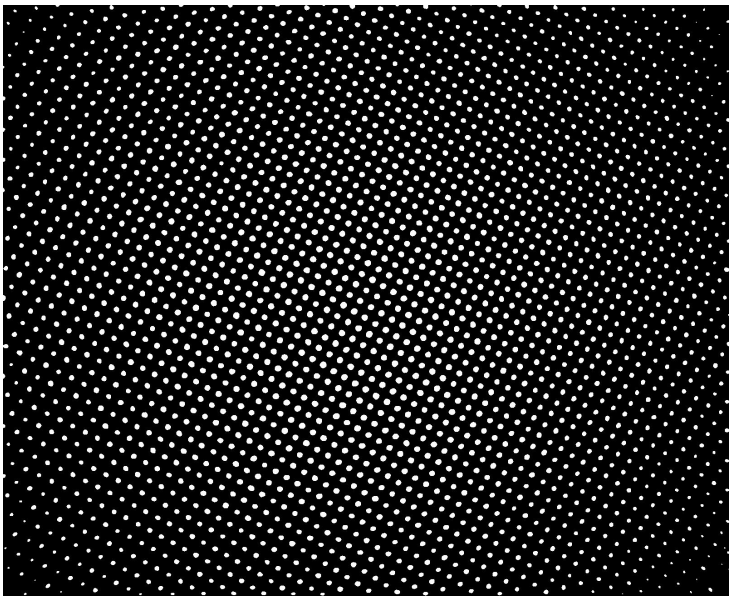


Figure 3.7: Calibration image after thresholding with a fix threshold has been applied.

To determine which threshold value to use for a large set of images was difficult since dot intensities could vary depending on imaged objects and light conditions of

the scene. Hence, it was useful to implement adaptive thresholding which found a suitable threshold value depending on the intensity levels of each frame. The Otsu method[29] was implemented to perform automatic adaptive thresholding, which found a suitable threshold by maximizing the variance between the set of values below the threshold and the set of values above the threshold. Using automatic thresholding was necessary when processing a large set of data, though in some cases manually chosen values could result in a better thresholding than what was obtained from the automatic algorithm. For instance, a too low threshold could result in clustering of dots which in turn produced outliers. In Figure 3.8 the thresholded image directly after Otsu thresholding is shown where the clustering of dots can be noted.

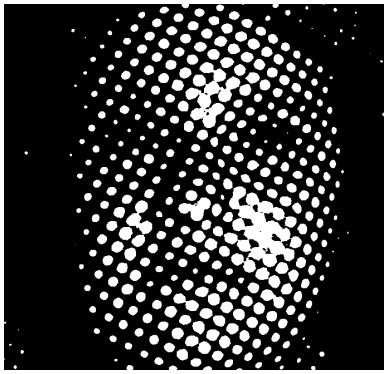


Figure 3.8: Thresholded image after Otsu thresholding, corresponding to the image in figure 3.2.

If the threshold was chosen manually, a higher threshold could be set to eliminate the clustering of dots, but this also resulted in loss of detected dots as the intensity of some projected dots was lower than the threshold value. An example of a case where the thresholded image is highly effected by relatively small changes in threshold is shown in Figure 3.9 and Figure 3.10.

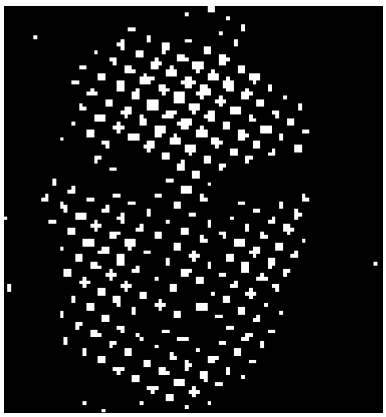


Figure 3.9: Thresholded image with threshold 19. The figure corresponds to the same original image as figure 3.10.

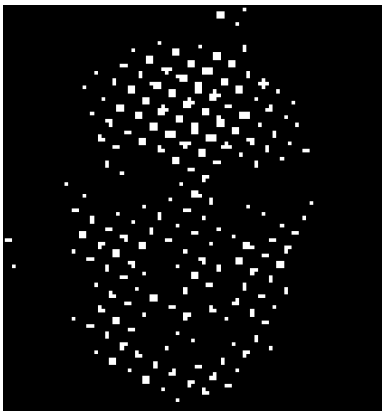


Figure 3.10: Thresholded image with threshold 20. The figure corresponds to the same original image as figure 3.9.

It is clear from these figures that it was not easy to set an optimal thresholding even when doing so manually, since the thresholded image often was highly affected by small changes in threshold. Usually, a higher threshold was preferable, as this reduced the number of outliers. Thereby, even though the automatically chosen thresholding was necessary for a large set of data, there were difficulties with using only automatic thresholding since it was not set high enough to enable an effective thresholding in all cases. Because of this, once an initial threshold was found using the Otsu method, an iterative increase of the threshold was implemented if the thresholded image was not considered to be processed well enough. The criteria for increasing the threshold was set depending on the area in pixels of the largest dot in the image. The dot area limit was set to be 1000 pixels and if exceeded the iteration was started. In each step of the iteration the threshold was increased by 5 and the maximum number of iteration was set to 6, due to limitations in time complexity. If the largest dot in the image still exceeded the maximum area of 1000 pixels the thresholding was considered not accurate, wherefore the image was neglected. The resulting thresholded image after this iterative updating of threshold can be seen in Figure 3.11. This image does not contain any clustering of dots, which the thresholded image before the iterative increase of thresholding in Figure 3.8 did. Instead, the dots in Figure 3.11 are all similar in size and can more easily be detected as dots by a blob detection algorithm.

Furthermore, in the process of thresholding the reduction of noise achieved by the Gaussian filtering can be seen more clearly, which can be seen in Figure 3.11 and Figure 3.12.

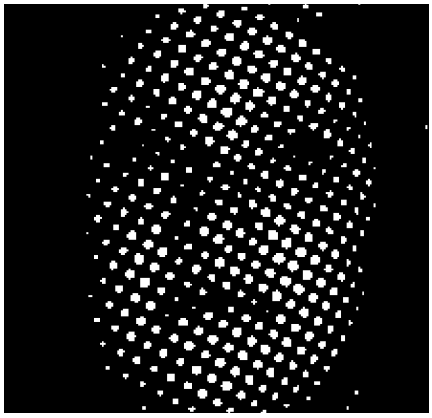


Figure 3.11: Final thresholded image with Gaussian filtering, corresponding to the image in Figure 3.3

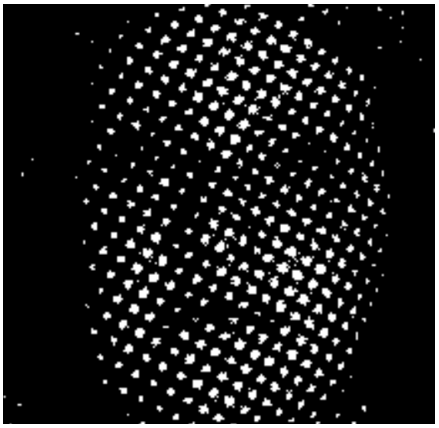


Figure 3.12: Final thresholded image without Gaussian filtering, corresponding to the image in Figure 3.3

Once thresholded images were obtained, a blob detector algorithm provided by OpenCV was used to find the centre of each thresholded dot. In the blob detector, the contour of each thresholded dot is found using edge detection algorithms. Then, weighted averages of pixel intensities, called image moments [30], corresponding to each thresholded dot is used to find properties such as centroid and area of the dot

corresponding to each contour. Image moments are calculated by:

$$M_{i,j} = \sum_x \sum_y x^i y^j I_{x,y}$$

where $I_{x,y}$ is the intensity at pixel position (x,y). The coordinates C_x and C_y of a centroid is defined using moments according to the following formula:

$$[C_x, C_y] = \left[\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right]$$

As the detection of dots required preprocessed images, centroids of dots were found on images which had been filtered with a Gaussian filter. The averaging of intensities caused by the Gaussian filter can reduce the accuracy of the estimation of centroids.

3.2.3 Estimation of depth

When the centres of dots were found the subsequent step was to match the dots between the image and the calibration image. For each dot in the image of the projected pattern a corresponding dot was searched for in the calibration image. An interval was set in the image, defined by the position of each dot in the image, and if only one dot was detected within the interval it was considered a match. If more than one dot were detected there was no matching, as no robust implementation could be designed to choose which dot to consider as a correspondence and which to dismiss. Each pair of corresponding dots detected was used to estimate disparity.

Once disparity had been found for each pair of corresponding dots in the image and the calibration image, the final task was to use the obtained disparity to estimate depth, using Equation 2.4 in Section 2.1.4. A depth map was constructed, where the estimated depth value was inserted at the pixel position for each matched dot in the image.

Once the depth estimation was completed interpolation was required to form a less sparse point cloud, resulting in a continuous depth map. Before interpolating the depth values, the values were masked to remove possible outliers as these have large impact on the interpolation. All values which deviated more than 10% from the median of the depth values were considered outliers, wherefore they were neglected. Cubic interpolation was implemented to create a dense depth map without sharp edges, in order to model the smoothness of the surface of the face. Once the final depth map was obtained a point cloud representation in 3D of the face was created for visualization.

In conclusion, the steps to achieve a continuous depth map were:

- Preprocessing of images and thresholding
- Detection of dots using a blob detector provided by OpenCV
- Matching of dots
- Computing disparity
- Converting disparity to depth and form depth map
- Interpolation of depth map

Lastly, the distance to the same position in space as estimated by the tracking algorithms had to be found in order to improve the current single point depth

estimate. The tracking algorithm estimates the distance to a position in between the eyes, which is situated behind the surface of the face. The corresponding 2D position is illustrated in red in Figure 3.13.

Figure 3.13: IR image where the 2D position of the single point depth estimate used in the evaluation process is illustrated in red.

The computed depth at this pixel was extracted from the depth map. Since the depth map models the distance to the surface of the face the measure is obtained by simply adding 1 cm, according to the limitations set for the project as specified in Section 1.4. This computed depth value was then compared to the estimated depth value obtained from tracking. Furthermore, the computed depth value was used as ground truth for networks in the second part of the project.

3.2.4 Evaluation of depth estimation

As the output of the depth estimation algorithm was to be compared with a single point depth estimate of the face provided by present software at Smart Eye the system had to be evaluated on human faces. However, when measuring distances on humans, errors might be introduced not only due to movements but also due to difficulties in measuring. In the evaluation process of the algorithm for estimating depth two test persons were included. In total 30 frames were captured for each person, at five different distances between 50 and 75 cm. The frame rate for each pair of images was 30 fps, wherefore the test person had to be still for one second at each test distance.

To reduce movements of the head a rack was used to position the head, which is shown in Figure 3.14. The forehead was put as far as possible into the upper part of the rack while the chest was put against the soft lower part of the rack. In Figure 3.15 an image of a test person positioned in the rack can be viewed and the corresponding image of the projected pattern is presented in Figure 3.16. By retaining the same pressure on the forehead and keeping all settings constant, the person was assumed to be still while taking images used for evaluation. Both parts of the rack can be adjusted horizontally and vertically to ensure a good fit for each person involved in collecting images.

Figure 3.14: Rack used to stabilize the head for collection of data used in the evaluation of depth estimation.

Figure 3.15: Test person positioned in the rack for depth evaluation.

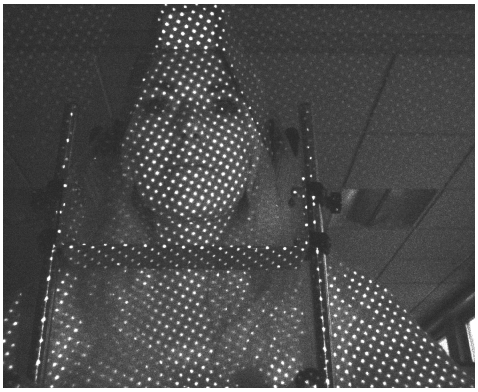


Figure 3.16: Corresponding image of the projected pattern for the person positioned in the rack for depth evaluation. The brightness of the image has been increased.

The distance between the camera and a position in between the eye on the bridge of the nose, as shown in Figure 3.13, was measured using a measuring tape. This position was chosen because it is the same position in 2D as the single point depth estimate obtained from the tracking as described in Section 3.2.3. As the reference

position was near the eyes the method for measuring control values had to be non-harmful to the eyes at all distances, which motivated the usage of a measuring tape.

Once all measures were obtained the mean value and standard deviation were calculated for each set of data, which is presented in Section 4.2.1 in Results. The objective of this evaluation was to evaluate precision, accuracy and invariance to characteristics of the head.

Lastly, the depth maps were evaluated more in detail. Here, the rack was not used as the objective was to evaluate the visual appearances of the depth maps, in both 2D and 3D. Different cases were studied, all in the same environment but with different distances and angles of the head, to find both successful and problematic cases.

3.3 Depth and pose prediction using depth maps

Once the performance of the depth estimation algorithm had been evaluated the second part of the project was initiated. The objective was to evaluate whether using depth maps computed by structured light improved the performance of an implemented depth and pose estimation network. Hence, a comparison was made between using only IR images as input and using IR images fused with depth maps as inputs. For both implementations, the same ground truth was used. The xy-position, rotation and facial landmarks were obtained from the tracking system whilst depth was extracted from the corresponding depth map. The comparison was performed on both a multitask network and a singletask network. The multitask network was implemented to predict rotation, xy-position, depth and landmarks while the singletask network was implemented to predict either depth or rotation. Depth maps were added using early fusion of inputs. In this section, the required steps to achieve the described comparisons are described in detail.

3.3.1 Collection of data

The objective of the data collection was to collect different head positions at different distances for a diversity of people. To achieve this a total of 11 test persons were included in the data collection, with a difference in characteristics, age and gender. No accessories such as glasses were included. All data were gathered at the same place in an office environment during all recordings. This was also the same place as where the calibration image for the pattern was taken.

The recording time for each test person was in total ten minutes, where eight minutes were primarily used for training and two minutes primarily for evaluation. For each person training data were recorded first and evaluation data were recorded directly after. In order to achieve different poses and distances instructions were set beforehand which the test person should follow. The different movements where divided into two different parts. Part A consisted of movements at given distances. The test persons did part A at a few different distances between approximately 40 and 75 cm distance from the camera. In the other part, part B, the movements were performed whilst moving towards and away from the camera. In this way many different distances were gathered per pose. The test persons were also asked to yawn

and laugh while looking around as a separate task. Both part A and part B were performed with both a neutral and a happy expression. Moreover, the test persons were encouraged to talk during the entire recording, in order to obtain different facial expressions in the different positions and distances. A few different poses and expression for one test person can be seen in Figure 3.17.

3.3.2 Generation of training and evaluation data

The framework for estimating pose and depth had to be adapted to generate new training and evaluation data with depth maps included as input. The tracking algorithms required to obtain parts of the ground truth was applied on IR frames. In order to generate training data the code was adapted such that output from the tracking algorithms was computed on IR frames, whilst depth maps and improved depth values were computed on the corresponding frames with projected structured light. As a result, a set of IR frame, depth map and output from tracking algorithms was generated for each frame and saved as training and evaluation data.

An IR frame was only saved as a new training example if it differed from previous frames. The examples were augmented with the same rotation, scale and translation for both the IR image and the depth map. For the IR image noise was added as well. For each frame that was estimated to differ enough from previous frames ten different augmentations were used. In total 277 954 examples for training were obtained and 53 752 for evaluation. At first, training was performed using a smaller data set, where each frame only was augmented once. As the resulting loss functions were not monotonically decreasing more augmentation was added to generate more data.

In order to evaluate the training data a subset of training examples was studied. In the visualization used for data evaluation, a 2D representation of the depth map as well as the corresponding IR image can be seen. A subset of such training data is presented in Figure 3.17. The background of the depth map is set to 50 cm during the visualization, in order to enhance the contrast in the images. For depth maps which are less than 50 cm the background will be white and for images where the depth is around 50 cm the background will be gray.

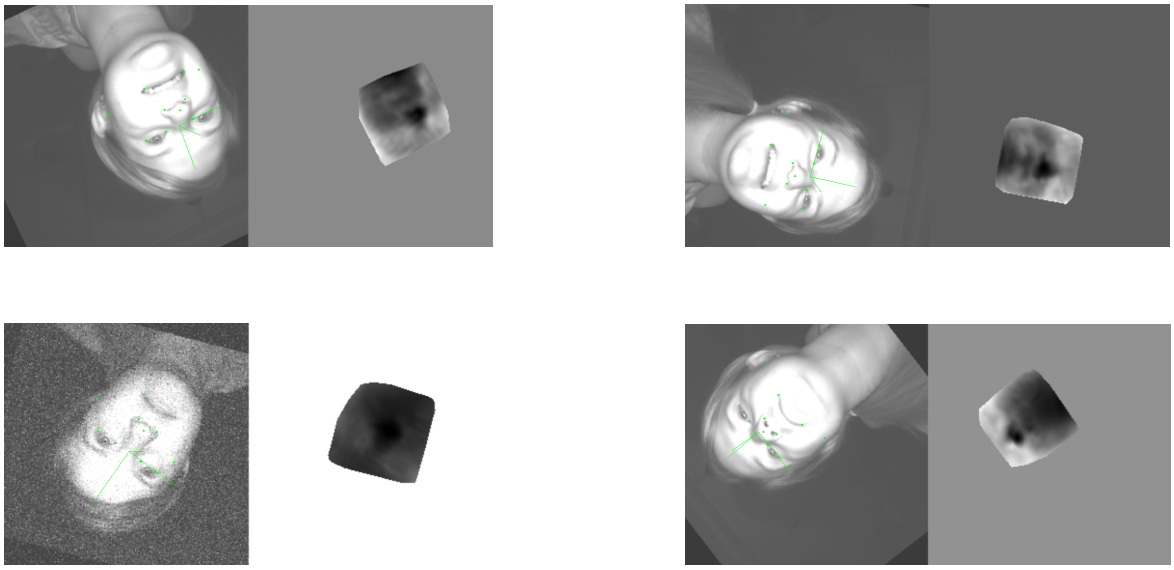


Figure 3.17: Training data showing the inputs to the network, i.e. an IR image and the corresponding depth map visualized in 2D.

3.3.3 Training and evaluating the networks

In order to study the impact of adding depth maps as input to CNNs a comparison was made between the performance of a network implementing only IR images as input and a corresponding network implementing both IR images and depth maps as inputs. For all cases of training and evaluation, the networks used were FAb-Nets with the optimizer Adam. The loss function was set to be the mean absolute error, MAE, and the networks were trained from scratch with random initial weights.

The loss function for rotation was computed using the ground truth of quaternions $\hat{\boldsymbol{q}}_{\text{gt}}$ and the predicted quaternion $\hat{\boldsymbol{q}}_{\text{pred}}$:

$$\hat{\boldsymbol{q}}_{\text{pred}}\hat{\boldsymbol{q}}_{\text{gt}}^{-1} = \hat{\boldsymbol{q}}_{\text{diff}}$$

From $\hat{\boldsymbol{q}}_{\text{diff}}$ the angular difference θ_{diff} was found:

$$\theta_{\text{diff}} = 2\arccos(w)$$

where θ_{diff} was minimized using MAE. This implementation of loss for rotation was already implemented in the available framework. For all other predicted parameters the loss function was trivially defined as the MAE of the predicted value as compared to the true value. As the training was not deterministic each case was trained twice to enable a more robust comparison between networks. The multitask network and single task network predicting depth were trained for 4000 epochs in total whilst the network for rotation was trained for 2000 epochs. The number of epochs was chosen depending on convergence and time limitations. For instance, training the multitask network for 4000 epochs required approximately 120 hours.

When depth maps were included as input early fusion was applied. Thus, the depth map was included as an additional channel in the input, which is clarified in Figure 3.18.

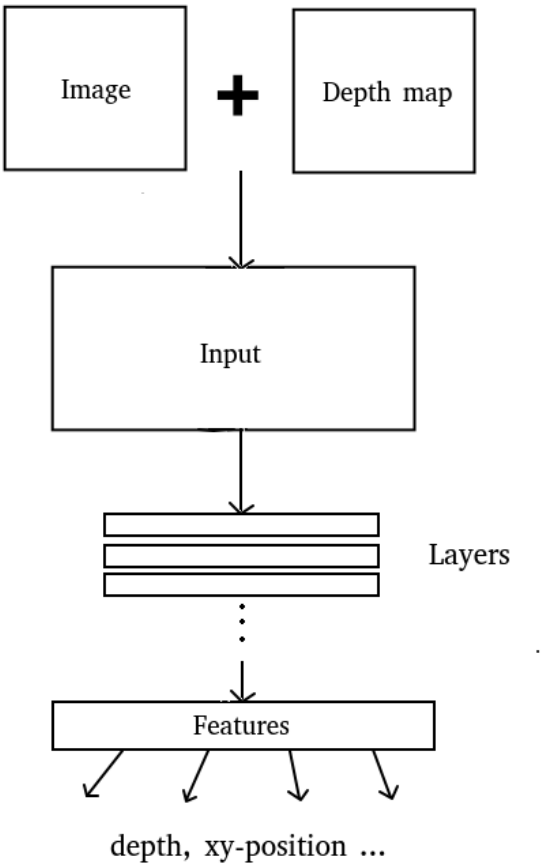


Figure 3.18: Illustration of early fusion used in this project. The symbol for addition corresponds to concatenation.

For testing, a late fusion network was trained separately on images and depth maps and the features from each training were concatenated as a final step. Then, outputs such as position and rotation were predicted. In Figure 3.19 a clarification of the structure is presented.

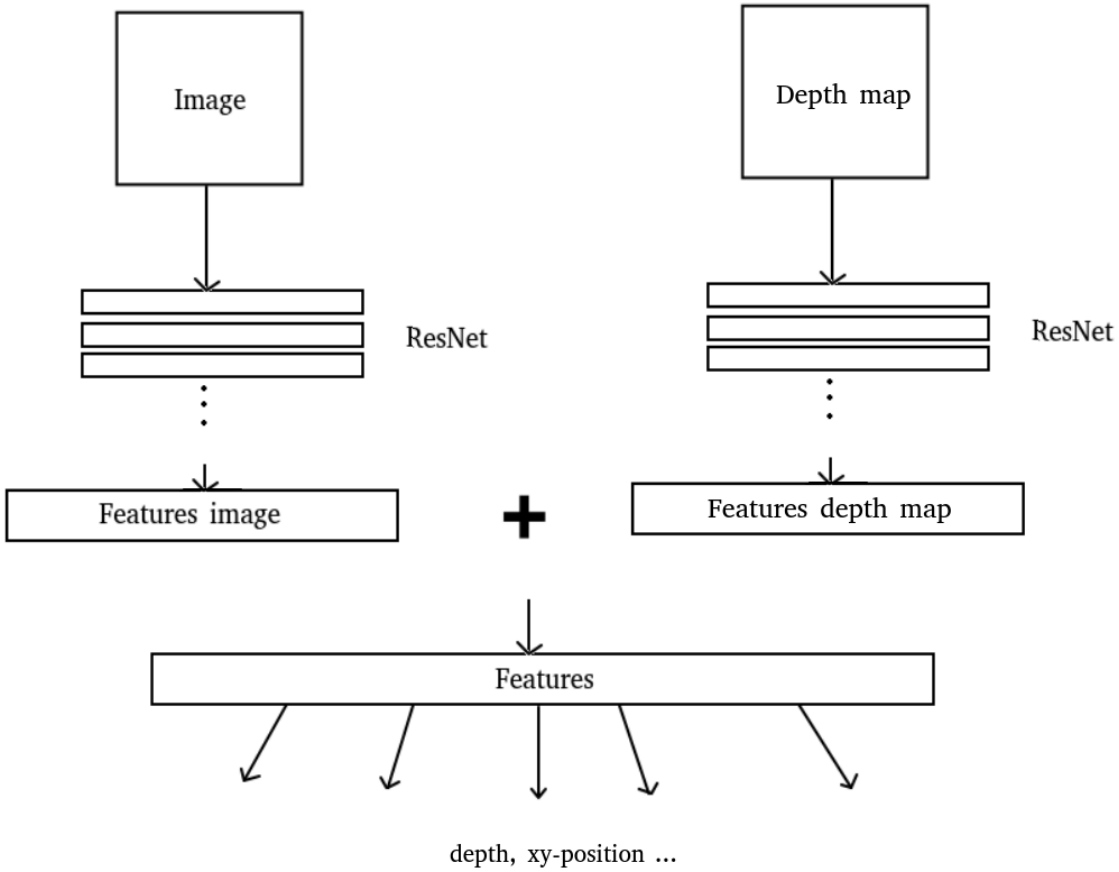


Figure 3.19: Illustration of late fusion. The symbol for addition corresponds to concatenation.

As the network was trained on an IR image and a depth map separately, late fusion was excluded in the final testing due to its time complexity. The time duration for training with late fusion was twice as long as training using early fusion.

Initially a ResNet was implemented and evaluated. However, this had a long convergence time wherefore a smaller network, a FAb-Net, was tried instead. In the final evaluation only FAb-Nets were evaluated as these were faster and resulted in monotonically decreasing loss functions.

Furthermore, both singletask networks and multitask networks were tested, as multitask networks have been found successful for face recognition tasks in previous studies [28]. The singletask network was set to predict either depth or rotation while the multitask network was set to predict depth, rotation, landmarks and xy-position of the single point depth estimate. For the multitask network all parameters were set to have the same priority.

4
Results

The following sections serve to present results obtained for both parts of the project. First, a theoretical analysis of the possibility to achieve set goals regarding the accuracy of the depth estimate is presented. Secondly, results obtained by the evaluation process of the depth estimation algorithm are presented as well as examples of generated depth maps. Finally, the performance of different implementations of networks are evaluated in order to be able to draw conclusion of the impact of adding depth maps as input to networks for prediction of depth and head pose, corresponding to the goal for the second part of the project.

4.1 Analysis of theoretical resolution

Before developing any algorithms for the prototype camera a theoretical analysis of the possibility to reach set goals with the specific pattern characteristics of the incorporated projector was carried out. The theoretical depth resolution of the Smart AI-X 3D was evaluated in the interval set for the project. The objective was to achieve a resolution higher than 1 cm. The analysis assumed points to be correctly matched between the pattern and the corresponding calibration pattern. Furthermore, the required resolution in disparity to achieve 1 cm resolution in depth at different distances was analyzed.

In order to analyze the resolution of a camera its specifics such as focal length, pixel size and baseline must be available. For the camera used for this project the focal length is 5.87 mm, the pixel size is 3 µm and the baseline of the device is 6 cm. In all further estimations the disparity resolution is assumed to be one pixel, i.e $\delta d = 1$. In Section 2.1.3 Equation 4.1 is obtained:

$$\delta D = \frac{D^2}{bf}\delta d \tag{4.1}$$

The objective is to investigate whether it is possible to obtain the requested depth accuracy of 1 cm given single pixel resolution in disparity at different distances. Hence, δD is set to be 0.01 m as this is what is asked for. To find which distances such depth resolution can be obtained for the formula in Equation 4.1 is rewritten as:

$$D = \sqrt{\frac{bf}{\delta d}\delta D} \tag{4.2}$$

Now, inserting the camera specific values in Equation 4.2 yields:

$$1.08 = \sqrt{\frac{0.06 \cdot 5.87 \cdot 10^{-3}}{3 \cdot 10^{-6}} \cdot 0.01} \tag{4.3}$$

According to the result above, given single pixel resolution in disparity it should be possible to obtain a resolution of 1 cm for distances up to 1.08 m.

Next, Equation 4.1 is evaluated for a set of distances D within the set interval and the results are presented in Table 4.1. By the result, it is clear that the theoretical accuracy in depth increases with distance but strictly stays below 1 cm for all distances of interest.

D [cm]	Resolution in D [cm]
30	0.08
50	0.21
70	0.42
100	0.85

Table 4.1: Theoretical depth resolution at different distances given single pixel resolution in disparity for the Smart AI-X 3D.

In all previous calculations a single pixel resolution has been assumed. Though, to find the pixel resolution necessary to detect a change of 1 cm in depth at different distances relative the calibration yet another analysis must be carried out. Equation 2.3 is applied assuming δD to be 0.01 m:

$$\delta d = \frac{bf}{D^2} \cdot 0.01$$

The difference in disparity caused by a change of ± 1 cm at different distances is estimated using the Equation above and the results are shown in Table 4.2. In order to be able to detect a change of 1 cm using an algorithm with single pixel resolution this difference must be larger than one pixel.

D [cm]	Δd for $\Delta D = 1$ cm [pxl]	Resolution of 1 cm in D [Y/N]
30	13	Y
50	5	Y
70	2	Y
100	1	Y

Table 4.2: Results required to evaluate whether it is possible to obtain 1 cm accuracy in depth at different distances relative the position of the Smart AI-X 3D.

By the result in Table 4.2 it is clear that all disparities are larger or equal to 1 pixel for set distances, wherefore sub pixel resolution is not necessary in the algorithm. It is also clear that for short distances, such as 0.3 m, the resolution in pixels can be quite low. For 0.3 m in distance, the limit for having 1 cm in accuracy is 13 pixels. The pixel resolution can therefore be off with 12 pixels and still be within the requested ± 1 cm interval at 0.3 m.

4.2 Results of depth estimation

The following sections serve to present results obtained regarding the estimation of depth. Once algorithms for estimating depth using structured light were implemented, a test was designed with the aim of evaluating the performance of these estimations. The accuracy of the depth estimation was evaluated on human faces and the result was compared with the requested accuracy of less than 1 cm in the interval 0.3 m to 1 m. Further descriptions of these tests and results obtained are presented in Section 3.2.4.

4.2.1 Evaluation of depth estimation

The depth estimation was evaluated according to the descriptions in Section 3.2.4 in the Methodology chapter. The test was executed on two test persons, at four different distances each with a total of 30 frames per distance and person. All results obtained from the testing are presented in Appendix.

Table 4.3 and Table 4.4 show the mean of the estimated distances obtained using the algorithm for depth estimation (A) along with the corresponding values obtained when using the present depth tracking system (T). Table 4.3 presents values obtained when using images of the first test person while Table 4.4 presents values obtained for images of the second test person. When comparing the mean values of the two tables it is clear that the accuracy of the current depth tracking algorithm at Smart Eye varies with person while the depth tracking algorithm does not. Table 4.5 presents standard deviation for both test persons. The system is precise, as the standard deviation never exceeds 7.5 mm. It can also be noted that the mean value of the depth tracking algorithm stays within one centimeter from the true value for all test cases. Moreover, 236 out of 240 tests stayed within one centimeter from the measured distance and the tests that did not are all taken at 50 cm for test person 2.

Distance [m]	Mean [m]	
	A	T
0.50	0.4926	0.5600
0.60	0.5978	0.6753
0.70	0.7044	0.7835
0.75	0.7492	0.8344

Table 4.3: Mean of estimated distances using the algorithm for depth estimation (A) and the depth tracking system (T) for test person 1 at four different distances.

Distance [m]	Mean [m]	
	A	T
0.50	0.4996	0.4618
0.60	0.5972	0.5778
0.70	0.7026	0.6685
0.75	0.7488	0.7200

Table 4.4: Mean of estimated distances using the algorithm for depth estimation (A) and the depth tracking system (T) for test person 2 at four different distances.

Distance [m]	SD [m]	
	Person 1	Person 2
0.50	$7.9{\cdot}10^{-4}$	$7.5 \cdot 10^{-3}$
0.60	$1.2{\cdot}10^{-3}$	$1.4 \cdot 10^{-3}$
0.70	$1.9{\cdot}10^{-3}$	$1.1 \cdot 10^{-3}$
0.75	$1.4{\cdot}10^{-3}$	$1.2 \cdot 10^{-3}$

Table 4.5: Standard deviation for measurements at four distances for two different test persons.

4.2.2 Generated depth maps

This section is divided to focus on a few different cases. First, the result of the example in Section 3.2 is visualized, along with another image taken from the same recording. Next, results for images in profile, images close to the camera and images captured far from the camera are presented. For each example the resulting depth maps are visualized in both 3D and 2D. The 3D representations is meant to visualize how well the depth map captures the shapes in the face, related to smoothness. In order to see the difference in 3D more clearly, a colormap is used. The colorbar shows the values in unit meter. The 2D representation shows how the depth map is input to the CNNs in Part 2 of the project. The face is positioned at the true position as described in Section 3.2.3 . The background of the depth map is set to zero when used as an input to the convolutional networks. Though, in the figures presented in this section the background is set to a value close to the rest of the values in the image in order to increase the contrast of the visualization of the face. In this section, the brightness has been enhanced for all images of projected patterns.

The resulting depth maps from the example discussed in the method, Figure 4.1 and Figure 4.2, are shown in 3D and 2D view in Figure 4.3, Figure 4.4 and Figure 4.5. The resulting point cloud is relatively smooth and the nose and lips can be seen in the the side view.

Figure 4.1: IR image of a person.

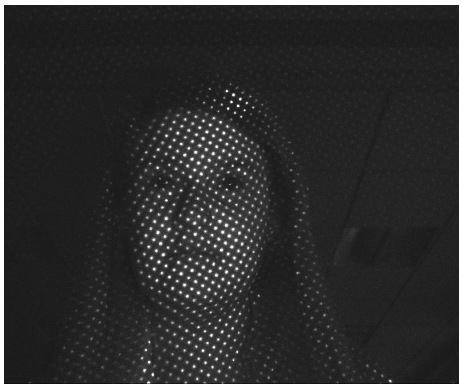


Figure 4.2: IR image of the projected pattern on a person, corresponding to the IR image in figure 4.1. The brightness of the image has been increased for visualization.

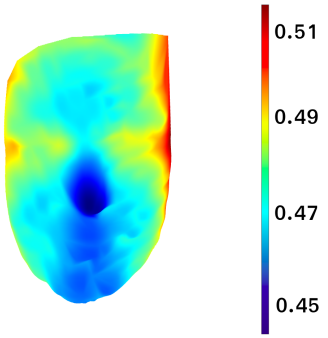


Figure 4.3: Resulting point cloud in front view when using Figure 4.1 and Figure 4.2 as input to the algorithms for depth estimation.

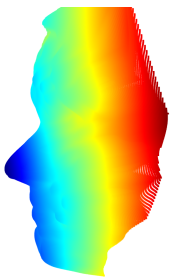


Figure 4.4: Resulting point cloud in side view view when using Figure 4.1 and Figure 4.2 as input to the algorithms for depth estimation.

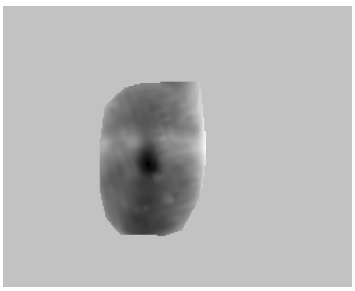


Figure 4.5: Resulting depth map view when using Figure 4.1 and Figure 4.2 as input to the algorithms for depth estimation.

A few seconds after the previous discussed image was taken the images shown in Figure 4.6 and Figure 4.7 was captured. It can be noted that the images differ from the previous images, but the intensity and distance to the person in both images is fairly similar. The resulting depth maps are different though, which can be seen in Figure 4.8, Figure 4.9 and Figure 4.10. Comparing Figure 4.4 and Figure 4.9, the two images in side view, the difference between the two depth maps is noted. The shape of the face in profile is not as smooth and does not follow the actual shape as well in this example, especially the nose has gotten very sharp and a shape which differ much from reality.

Figure 4.6: IR image of a person, obtained a few seconds after Figure 4.1.

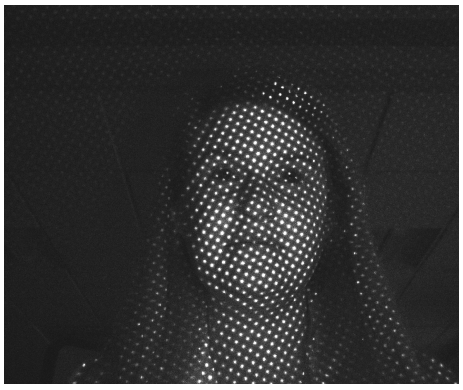


Figure 4.7: IR image of the projected pattern on a person, corresponding to the IR image in Figure 4.6. The brightness of the image has been increased for visualization.

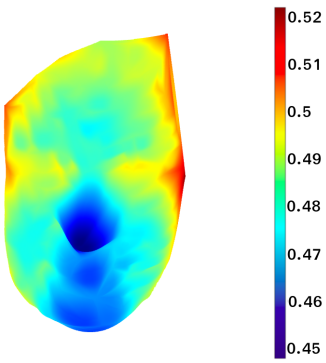


Figure 4.8: Resulting point cloud in front view when using Figure 4.6 and Figure 4.7 as input to the algorithms for depth estimation.

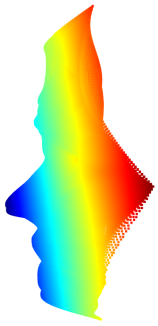


Figure 4.9: Resulting point cloud in side view when using Figure 4.6 and Figure 4.7 as input to the algorithms for depth estimation.

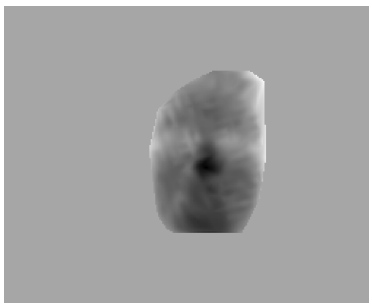


Figure 4.10: Resulting depth map when using Figure 4.6 and Figure 4.7 as input to the algorithms for depth estimation.

The next test case consider images of a person positioned almost in profile and is visualized in Figure 4.11 to Figure 4.14. It is clear that the lower part of the face is easier to represent than the upper part in this case, due to the upper part having a lower intensity.

Figure 4.11: IR image of a person positioned almost in profile.

Figure 4.12: IR image of the projected pattern on a person positioned almost in profile, corresponding to the IR image in Figure 4.11. The brightness of the image has been increased for visualization.

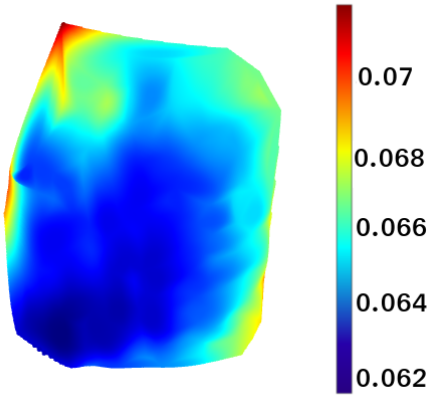


Figure 4.13: Resulting point cloud from obtained when using Figure 4.11 and figure 4.12 as input to the algorithms for depth estimation.

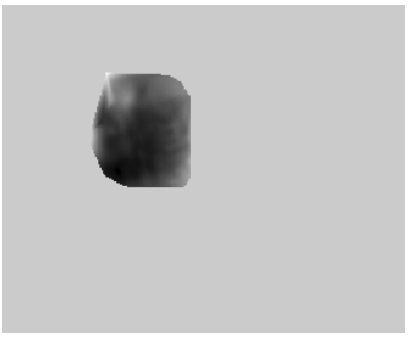


Figure 4.14: Resulting depth map obtained when using Figure 4.11 and Figure 4.12 as input to the algorithms for depth estimation.

A set of cases where outliers are present is illustrated in Figures 4.15 to 4.19. This image is captured at a distance closer to the camera, at 40 cm, than what the depth estimation is evaluated on. Hence, this case is included because of Part 2, since this distance occur in the data collection.

Figure 4.15: IR image of a person located close to the camera.

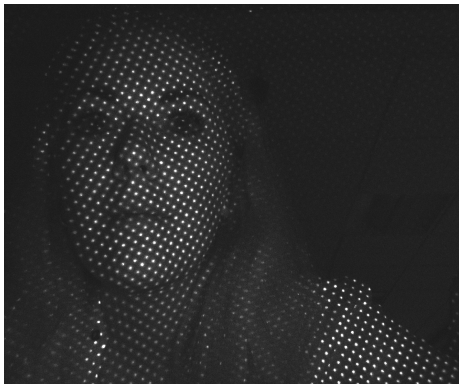


Figure 4.16: IR image of the projected pattern on a person located close to the camera, corresponding to the IR image in Figure 4.15. The brightness of the image has been increased for visualization.

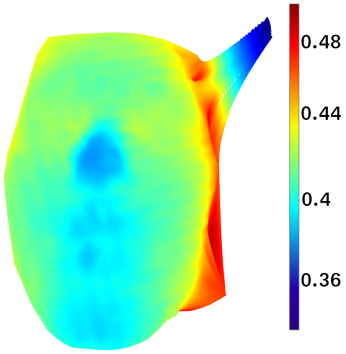


Figure 4.17: Resulting point cloud in front view when using Figure 4.15 and Figure 4.16 as input to the algorithms for depth estimation.

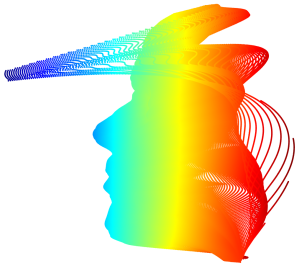


Figure 4.18: Resulting point cloud in side view when using Figure 4.15 and Figure 4.16 as input to the algorithms for depth estimation.

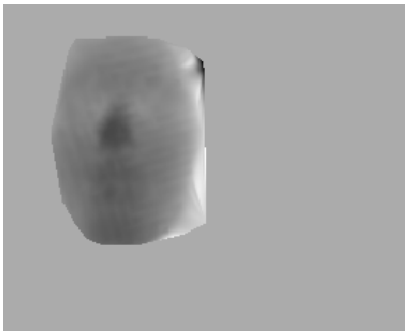


Figure 4.19: Resulting depth map when using Figure 4.15 and Figure 4.16 as input to the algorithms for depth estimation.

In Figure 4.20 to Figure 4.24 instead, a person is located far away from the camera. The distance is approximately 90 cm. It is clear from Figure 4.20 and Figure 4.21 that the intensities in the images are low, resulting in dark dots with low contrast. In turn, calculating depth from such dots is problematic and the point cloud obtained is not as smooth as for closer distances.

Figure 4.20: IR image of a person located far from the camera.

Figure 4.21: IR image of the projected pattern on a person located far from the camera, corresponding to the IR image in Figure 4.20. The brightness of the image has been increased for visualization.

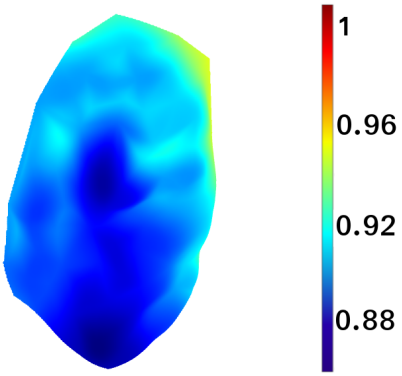


Figure 4.22: Resulting point cloud in front view when using Figure 4.20 and Figure 4.21 as input to the algorithms for depth estimation.

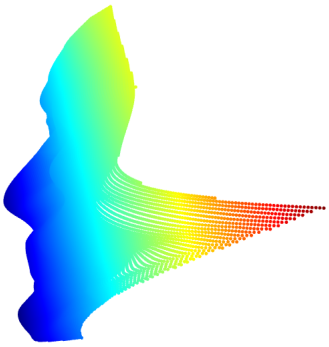


Figure 4.23: Resulting point cloud in side view when using Figure 4.20 and Figure 4.21 as input to the algorithms for depth estimation.

Figure 4.24: Resulting depth map when using Figure 4.20 and Figure 4.21 as input to the algorithms for depth estimation.

Figure 4.25 and Figure 4.27 show point clouds obtained when using Gaussian filtering in front view and in side view respectively, for the example discussed in the methodology. Figure 4.26 and Figure 4.28 show the same depth map but without using any Gaussian filter. When comparing the images it is clear that the filtering improves the result since it greatly reduces the level of noise.

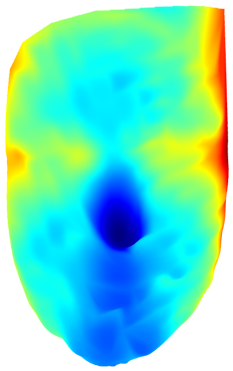


Figure 4.25: Obtained point cloud when using Gaussian filtering as preprocessing of images.

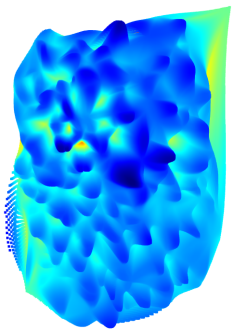


Figure 4.26: Obtained point cloud without using Gaussian filtering as preprocessing of images.

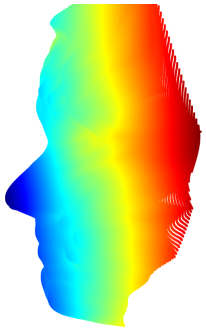


Figure 4.27: Side view of an obtained point cloud when using Gaussian filtering as preprocessing of images.

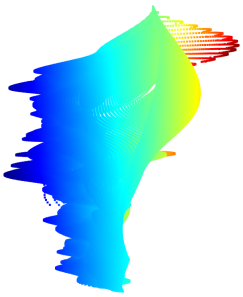


Figure 4.28: Side view of an obtained point cloud without using Gaussian filtering as preprocessing of images.

4.3 Results of depth and pose prediction

In the following sections the results obtained for the second part of the project are presented. The performance of a network when adding depth maps as additional input was compared to the case when only using IR images as input. Two different networks were implemented, both networks were based on the FAb-Net and early fusion of data when depth maps were included as input. In all other aspects the networks were identical. For all cases, mean absolute error was set as loss function. First, a singletask network which estimated depth and rotation separately was implemented. Later, a multitask network was implemented as well, predicting depth, rotation, facial landmarks and xy-position of the single point depth estimate. Each network was trained on the entire training set consisting of 277 954 frames and evaluated on the validation set consisting of 53 752 frames. Each network was trained twice, with random initialization of weights. For the single task networks predicting rotation and depth the training which resulted in the lowest loss value for the validation was chosen for the comparison. When using the multitask network, the training which resulted in the lowest total loss for validation, including all parameters, where chosen for the comparison. In all figures in this section, the evaluated loss functions for validation are plotted against time, as it was considered most convenient when combining multiple training sessions for visualization in TensorBoard. Hence, the flat line which can be observed in for example Figure 4.29 corresponds to the time passed between different training sessions.

4.3.1 Singletask network

Initially, a singletask network was set to predict only depth or rotation. In Figure 4.29 and Figure 4.30 the loss function is evaluated for the validation set when using IR images as input only and when adding depth maps to the input by early fusion for each case. In Figure 4.29 the green graph represents the case when depth maps are included as input while the pink graph only uses IR images as input. The loss function is simply the mean absolute error of the predicted depth. It is clear that adding depth maps as input to the network decreases the loss for prediction of depth and thereby increases the performance of the network. Next, in Figure 4.30 the orange graph represents the case when depth maps are included as input while the blue graph only uses IR images. The loss function for rotation is estimated using quaternions. By the figure, it can be observed that adding depth maps as input to the network decreases the loss for prediction of rotation and thereby increases the performance of the network. A close up on the last epochs is included in Figure 4.31.

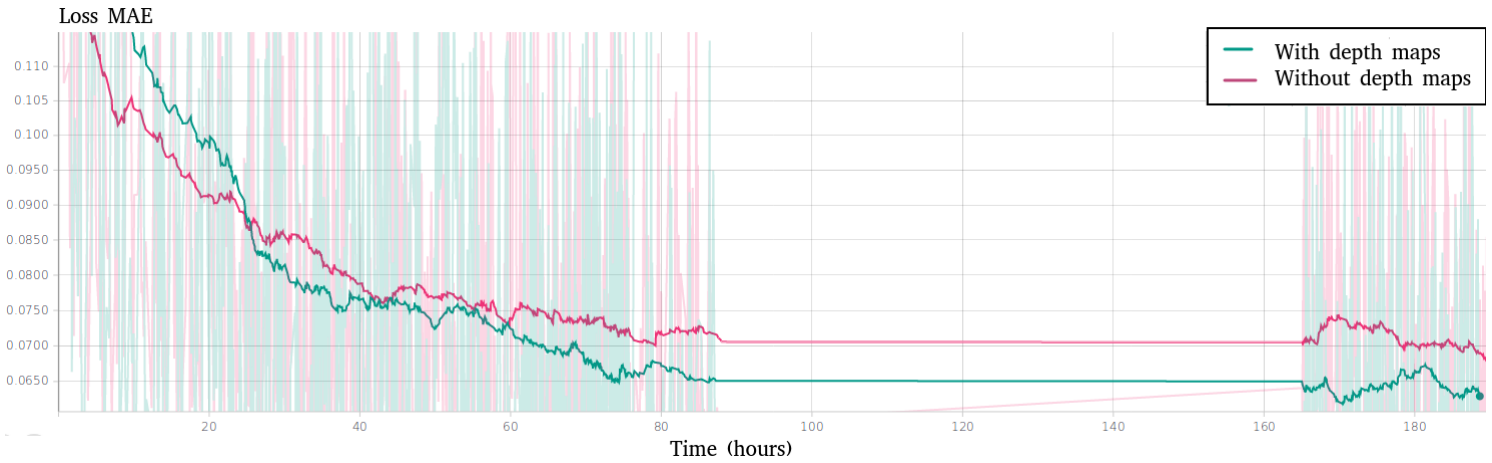


Figure 4.29: Loss function for prediction of **depth** evaluated on the validation set with and without adding depth maps as input to the singletask network.

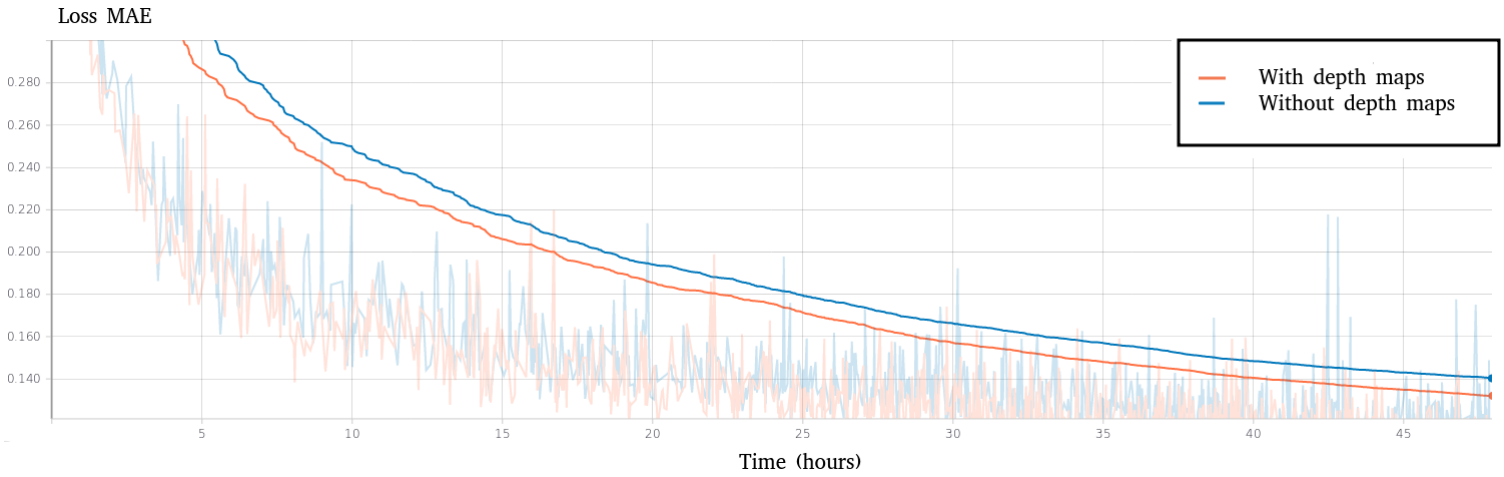


Figure 4.30: Loss function for prediction of **rotation** evaluated on the validation set with and without adding depth maps as input to the singletask network.

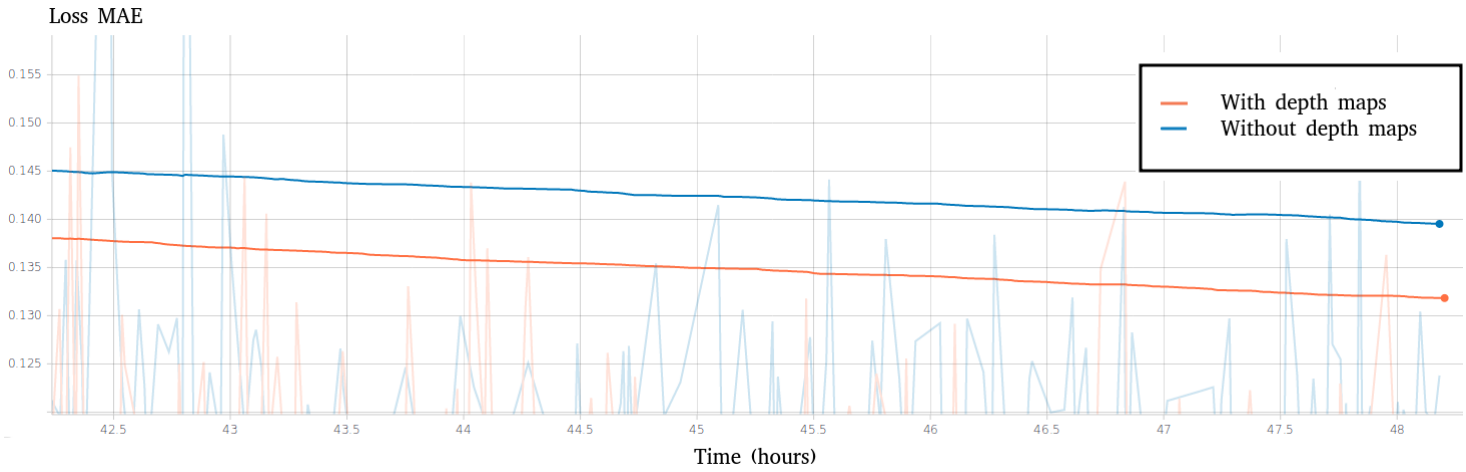


Figure 4.31: Zoom in on the loss function for prediction of **rotation** evaluated on the validation set with and without adding depth maps as input to the singletask network.

Furthermore, the loss in the very last epoch for each implementation is presented in Table 4.6, as well as the percentage decrease in loss when adding depth maps as input.

Loss	IR	IR + DM	Decrease [%]
Depth	0.0684	0.06294	8.0
Rotation	0.1225	0.1131	7.7

Table 4.6: Loss evaluated in the very last epoch for prediction of depth and rotation respectively, when using either IR images or both IR images and depth maps as inputs.

4.3.2 Multitask networks

As a second step, a multitask network was implemented to predict depth, rotation, facial landmarks and xy-position of the depth estimate. For each parameter the loss function is defined to be the mean absolute error of the predicted parameter. as for the singletask network the loss function for rotation uses quaternions. The results obtained are presented in Figure 4.32 to Figure 4.36, where the red graph and the light blue graph represent networks with and without adding depth maps as input. By Figure 4.34, Figure 4.35 and Figure 4.36 it is clear that adding depth maps as input to the multitask network significantly increases the performance of prediction of facial landmarks, xy-position and rotation. In all of these figures there is a clear distinction between the graphs representing adding depth maps and not adding depth maps, wherefore only the graph with the lowest loss for validation is presented. Though, for prediction of depth the performance when including depth maps differs between the two runs, wherefore all four runs are presented. The results for prediction of depths are presented in Figure 4.32 and Figure 4.33 where the light

blue graph and the green graph represent adding depth maps, while the red graph and the purple graph represent not using depth maps.

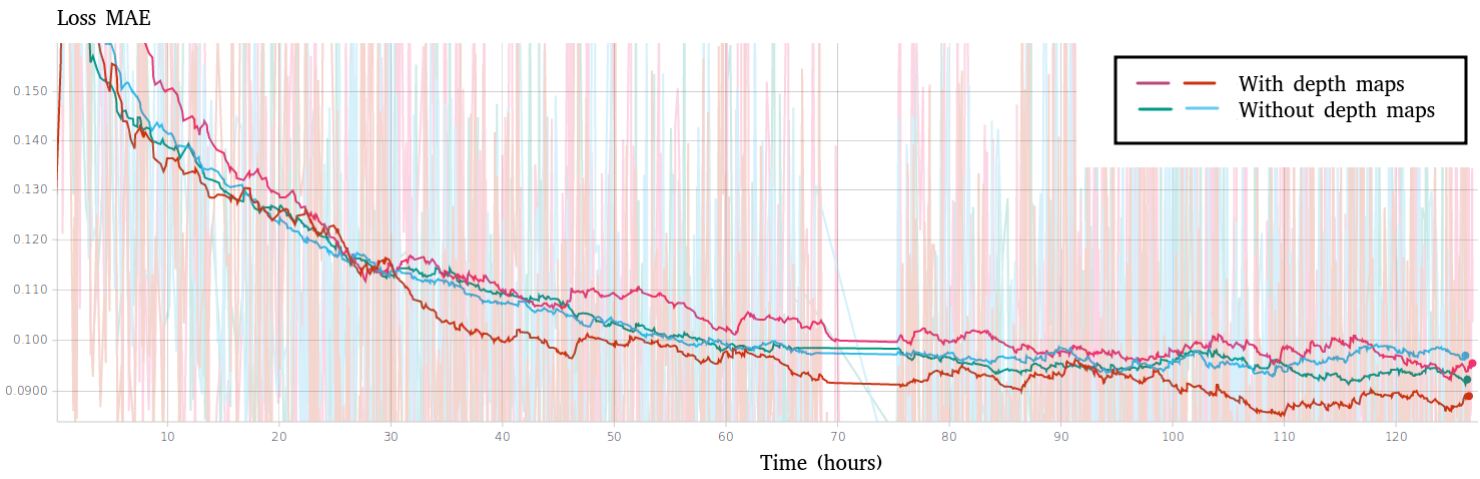


Figure 4.32: Loss function for prediction of **depth** evaluated on the validation set, both with and without adding depth maps as input to the multitask network.

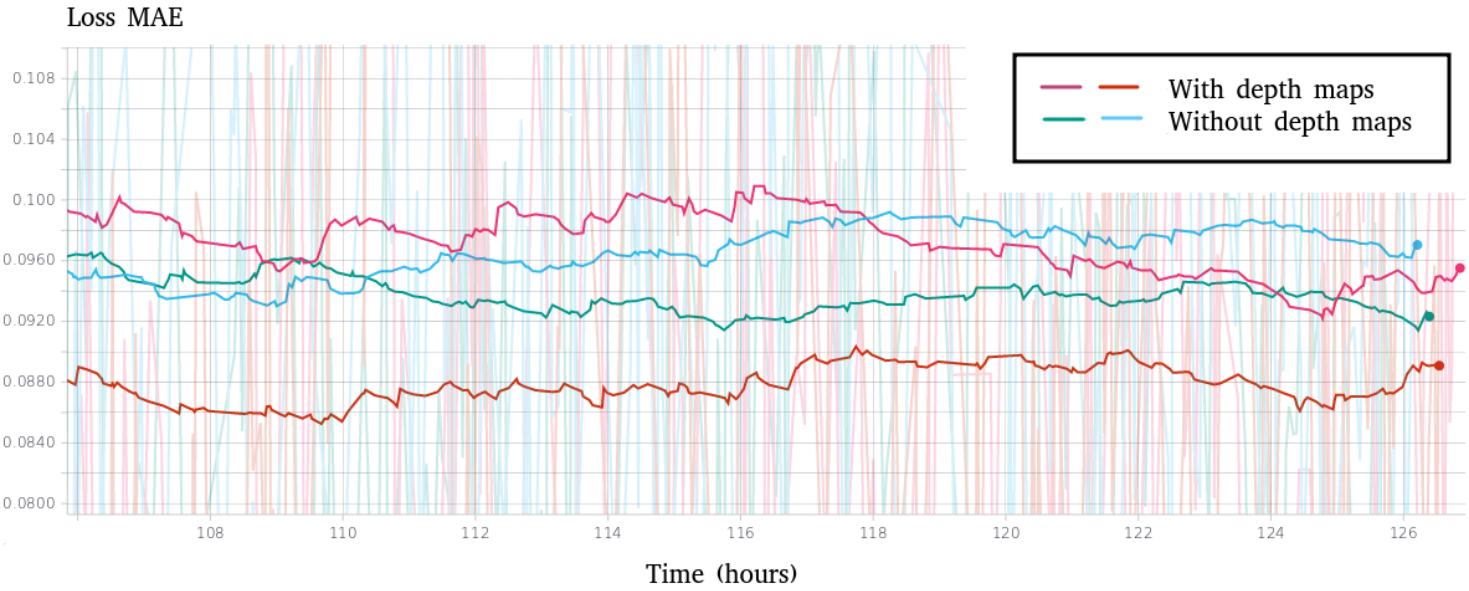


Figure 4.33: Loss function for prediction of **depth** evaluated on the validation set, both with and without adding depth maps as input to the multitask network.

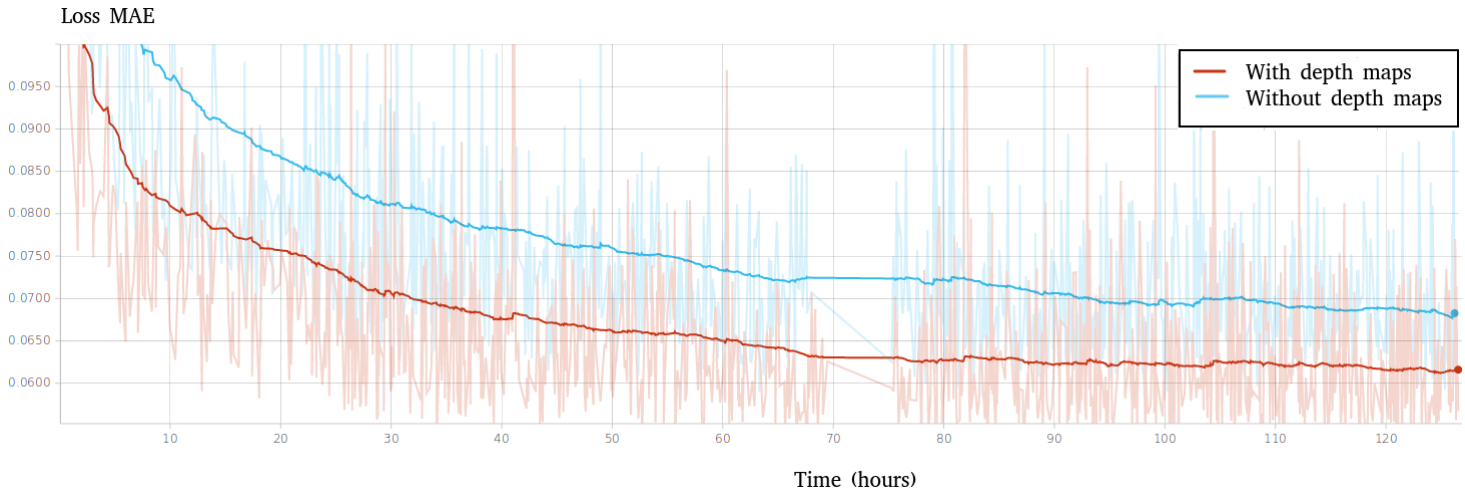


Figure 4.34: Loss function for prediction of **facial landmarks** evaluated on the validation set, both with and without adding depth maps as input to the multitask network.

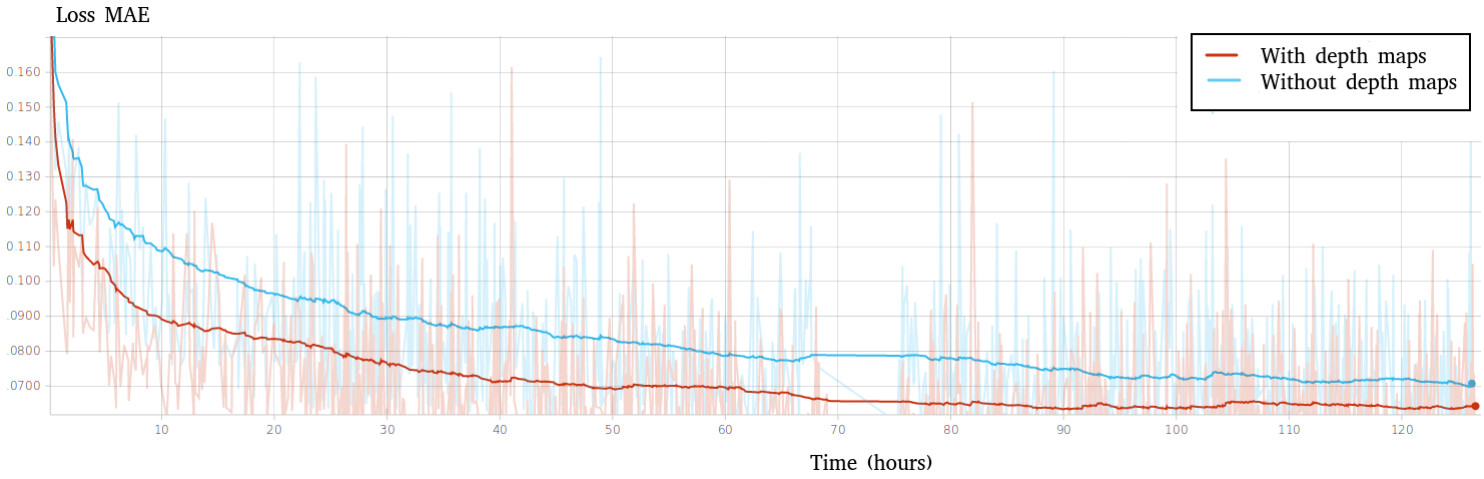


Figure 4.35: Loss function for prediction of **xy-position** evaluated on the validation set, both with and without adding depth maps as input to the multitask network.

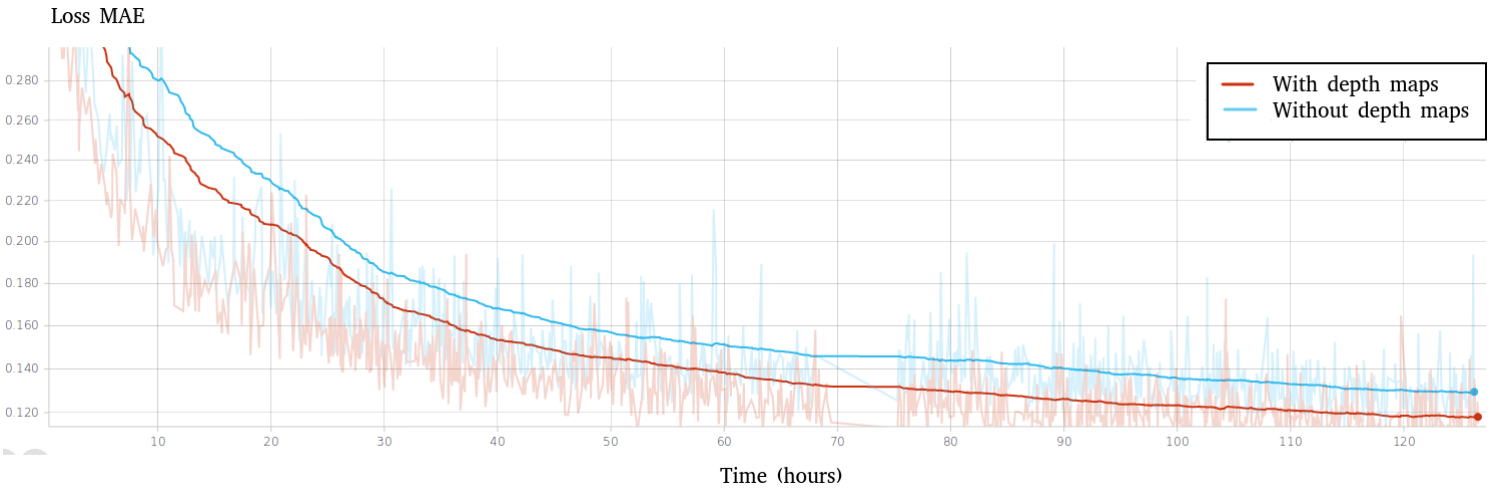


Figure 4.36: Loss function for prediction of **rotation** evaluated on the validation set, both with and without adding depth maps as input to the multitask network.

The losses in the very last epoch for prediction of rotation, xy-position and landmarks are presented in Table 4.7, as well as the percentage decrease in loss when adding depth maps as input. In Table 4.8 the losses in the very last epoch for prediction of depth are presented.

Loss	IR	IR + DM	Decrease [%]
Rotation	0.1295	0.1181	8.8
Xy-position	0.0707	0.06424	9.1
Landmarks	0.06827	0.06160	9.8

Table 4.7: Loss evaluated in the very last epoch for multitask prediction of rotation, xy-position and landmarks, when using either IR images or both IR images and depth maps as input. Depth maps are denoted by DM.

	Loss for prediction of depth
IR	0.09704 0.09232
IR + DM	0.09552 0.08908

Table 4.8: Loss evaluated in the very last epoch for multitask prediction of depth, when using either IR images or both IR images and depth maps as input. Depth maps are denoted by DM.

5
Discussion

In this chapter, the results are discussed in further detail. Furthermore, improvements as well as future steps for this project are presented.

5.1 Depth estimation

Using structured light for computation of depth maps resulted in well estimated depth and depth maps which visually looked accurate. Nevertheless, some difficulties related to the challenges mentioned in Section 2.1.5 can be further improved. In this section possible improvements of the algorithm, restrictions in evaluation and future steps of using structured light for depth estimation are discussed.

5.1.1 Summary of evaluation of depth estimation

In the process of evaluating the accuracy of the depth estimation algorithm a total of 240 measurements were taken, out of which only 4 were outside the requested interval of ± 1 cm. All of these four measurements were also obtained during the same recording. Hence, it is possible that something occurred during this recording which affected the accuracy, for instance small movements of the test person or unexpected behaviour of the projector. Furthermore, the four values were inside an interval of $\pm$ 2 cm, and the mean value for that recording was still within the requested interval. The exact accuracy of the system could not be set due to limitations in measuring of the true distance, but the evaluation can still be considered successful as the accuracy measured was within the requested interval.

The obtained accuracy is a great improvement compared to the current system which had an accuracy of around 8 cm for test person A and 3 cm for test person B for all four distances. As each person deviates more or less from the statistical head model used in the current system, each person will have different accuracy in the depth estimate obtained from tracking. Another goal with the depth estimation using structured light was invariance to head characteristics which was fulfilled since the accuracy of the computed depth value did not differ between the two test persons in the new system. The accuracy of the current system might be misleading, as described in section 1.1, the tracking algorithms demands a head model of the face which is created from rotation of the head which was not possible when using the rack for evaluation. The accuracy for the current system was hence included only as a reference in order to note the impact of characteristics on the current system compared to the new algorithm. In the new system no calibration of this sort is

needed in order to obtain the depth estimate which is an advantage.

The precision of the system was calculated in order to ensure that the new system did not suffer from problems related to precision. The precision was in the order of millimeters, which was considered high enough. In conclusion, using structured light for computing depth was successful with respect to both accuracy, precision and invariance to head characteristics.

5.1.2 Further improvements of depth estimation

Even though the depth estimation was considered successful, improvements could still be done. The two most distinct challenges in the proposed algorithm were the differences in light intensity in the images and occlusion. Occlusion is also one of the biggest issues related to structured light with only one camera, as mentioned in Section 2.1.5. Regions behind the nose were often a difficulty, which resulted in outliers. Most of these outliers were removed by masking before interpolation of depth maps, described in section 3.2.3. In order to handle the problem of occlusion the projector on the other side of the camera could be included in the system as well. Due to its position it would not result in the same occluded areas as the projector used in this project wherefore the detected dots from each projected pattern could be combined to fill holes caused by occlusion. Though, this would require yet another frame of projected pattern, resulting in a lower frame rate than what is currently implemented.

Regarding the differences in light intensity, the main difficulty was finding the optimal thresholding. A possible way to handle this would have been to apply different thresholding at different regions of the image. This method was tested, but did not yield improved results as the areas with very high intensity were most often relatively small. Therefore, an efficient division of the cropped image still yielded too large variations in intensity within some of the divided parts of the frame. As a result, this implementation was excluded and all thresholding was applied to the entire cropped image instead. Though, with the aim of increasing the accuracy a future step could be to fine tune the thresholding for each dot, so that each centroid could be detected more carefully. Clearly, this would increase the time complexity of the algorithm, which also would have to be considered.

Thresholding was important since a blob detector dependent on thresholding was used. In order to generate depth maps for the data set for Part 2, it was necessary that the blob detection was fast. The chosen blob detector seemed to have a reasonable accuracy along with high speed. Had low time complexity not been important other blob detectors could have been chosen instead, which were not as dependent on the thresholding.

Evaluating the performance of the blob detector was a difficult task as the contrast of each dot was low, which can be seen in Figure 3.6. Hence, whether the thresholding actually affected the accuracy of detection of dots is not clear. Due to low contrast, defining the true centroid of dots manually for comparison with the ones detected by the blob detector was not considered possible, which is why the blob detector was not evaluated further.

The noise level of the input image clearly affected the design of the system, which

was solved by applying the Gaussian filter. If the noise could be reduced, for instance by decreasing the gain of the camera, such filtering would not be necessary. Though, this would probably result in a lower contrast of dots, which in turn caused challenges as described above. Due to difficulties in defining the true centroid as described above, any algorithm to counteract the possible decrease in accuracy caused by the Gaussian filtering was not possible.

In research, pseudo random patterns are often used for structured light, as mentioned in Section 2.1.4 in Theory. If pseudorandom pattens were implemented the matching of dots as well as the blob detection would be excluded from the algorithm, since other pattern recognition algorithms could be applied instead. Since these processes caused most outliers it would be interesting to investigate if the performance would be improved by using pseudorandom patterns.

5.1.3 Restrictions and simplifications of the evaluation

The impact of ambient light was not studied during the project, since all data were collected in the same environment. Though, to make the system useful in a compartment such a study must be carried out to investigate how the system performs in common lightning conditions such as sunlight during the day and darkness during the night. Moreover, testing in a real compartment could exploit challenges caused by natural movements while driving. Hence, even though the system can be assumed to perform well in a controlled environment its usability in a real case scenario must be further investigated. Moreover, no glasses were worn during the evaluation nor the data collection due to the reflectivity of the glass. Though, it is obvious that such characteristics must be handled to make the system useful in a real scenario. Since the evaluation of the depth estimate only included two persons it would be of interest to include a larger set of persons in the evaluation, so that more characteristics could be tested as well.

As described in Section 3.2.4, a rack was used in the process of evaluating the accuracy of the depth estimate. This rack was constructed such that only images on the face in frontal position could be captured, at certain distances. For short distances the rack itself was in front of the person resulting in occlusion. Hence, short distances could not be evaluated. Furthermore, it was not possible to evaluate images in profile due to the construction of the rack. The depth maps for a limited amount of cases were studied visually, but no further evaluation was performed. However, cases which were not studied are present in Part 2 of the project. To ensure that only accurate depths were sent to the networks in the second part of the project an evaluation of more positions and distances would be required as well as an evaluation of the accuracy of the entire depth map. Though, in order to evaluate the entire depth map a true depth map must be available for comparison, which was not the case for this project.

The evaluation process could be expanded if simulated data were included. Simulating the projection of the pattern on a mesh of the human head would enable detailed evaluation of numerous aspects of the algorithm. For instance, the accuracy of the depth map for different position and distances to the head as well as different characteristics could be evaluated for a large amount of data with high accuracy if

the depth at each pixel was defined by the simulation. Furthermore, the individual steps in the algorithm could be evaluated which would be valuable as mentioned in Section 5.1.2.

Another restriction was the conversion between the position of the tracked distance and the position of the calculated distance using structured light. As mentioned in Section 1.4, the difference between these positions was limited to be set to 1 cm for all cases. Due to this limitation, the distances will not correspond to the exact same position if the real distance between those positions deviates from the set value of 1 cm.

5.1.4 Future work for Part 1

There are several aspects which would be of interest to investigate further regarding the first part of the project, as discussed in previous sections. In summary, the most important steps for future work would include:

- Evaluation of the system on more test cases including simulated data
- Enabling usage of both projectors of the device to avoid occlusion
- Investigation of whether a projected pseudo random pattern could increase the performance of the system

5.2 Prediction of depth and pose

In the following sections the results obtained when comparing the performance of a FAb-Net implemented to use IR images as input to a FAb-Net implemented to use both IR images and depth maps as inputs are discussed. Future steps regarding the usage of depth maps as input to networks are also presented.

5.2.1 Summary of results for prediction of depth and pose

When adding depth maps as input to a single task network for predicting either depth or rotation the performance was found to increase, as seen in Figure 4.29 and Figure 4.30. For prediction of rotation the performance was increased instantly while several epochs was needed before a clear difference could be observed for prediction of depth. In conclusion, adding depth maps as input for the singletask network resulted in improved performance for all evaluations of each task, wherefore it can be concluded that depth maps are beneficial for such predictions.

Moreover, from Table 4.6 it can be noted that the increment in performance is slightly larger for prediction of depth. Though, since the difference in increments is small it is necessary to train each network more than twice to be able to draw more robust conclusions. For this project, each network was set to only be trained twice as each training was very time consuming. Therefore, no comparison between the improvements for different parameters is carried out.

For the multitask network, adding depth maps as input was found to increase the performance regarding prediction of facial landmarks, xy-position of the single point depth estimate and rotation. Hence, the depth maps, which represent 3D information, served to increase the performance of two dimensional tasks as well.

For all three implementations, the increment in performance was observed from the very start of the training and consisted for all epochs. Again, the differences in increments presented in Table 4.7 are quite small, wherefore a larger number of trainings per network would be interesting to evaluate. Including depth maps for prediction of depth using the multitask network was ambiguous when compared to the best loss obtained when not using depth maps, as one training resulted in decreased loss while the other training resulted in increased loss. For this case, it would be interesting to train for an even larger number of epochs to see if any clear difference could be obtained. The comparison was performed using only one network architecture and it is possible that the result would be different if another network architecture was chosen.

When comparing the prediction of depth for the singletask network and the multitask network it can be observed that the prediction of depth was increased for the singletask network while no significant change in performance could be concluded regarding prediction of depth for the multitask network. The losses presented in Table 4.6 for prediction of depth are relatively small in comparison with the multitask losses for the same task, presented in Table 4.8. Therefore, it seems as the prediction of depth was not profited by including prediction of more features in the network. Though, it is clear by Figure 4.32 that the graphs have not converged. Therefore, it is still possible that a result closer to that obtained for the singletask network would be obtained if the training was to be continued for a larger number of epochs or if other training parameters had been implemented.

Furthermore, this project aimed to study the impact of adding depth maps as additional input to a network and did not consider optimizing the result for each network. Thus, it is left to investigate how changing network architecture or learning rate can affect the losses obtained.

5.2.2 Evaluating the data set

From the figures presented in Section 4.2.2 it can be noted that even if outliers appear obvious when representing the data as a 3D point cloud such outliers can be difficult to recognize in the depth map. For instance, in Figure 4.19 the outlier is not obvious whilst in Figure 4.18 the outlier is clear. Therefore, when examining the generated training data the depth maps can be interpreted as smooth even though the point cloud would reveal the existence of outliers. The visualization of the training data was made in 2D, as presented in Section 3.3.2. Hence, there is a risk that the depth maps included as inputs to networks contained outliers. It would be beneficial to be able to evaluate the data further in order to estimate the quality of the input.

Furthermore, in this project the pattern of the device is projected every other frame. Hence, using depth maps resulting from the projected pattern as a correspondence to IR images from a previous frame introduces an error if there are rapid movements in the scene. If such errors could decrease the performance of networks, due to introduction of noise, was not studied during the project.

5.2.3 Train on simulated data

The data collection was limited to include 11 persons whereas augmentation was used to increase the training and evaluation set. Obviously, it would be beneficial to include a larger set of data where more poses, expressions and a wider range of people were included.

A possible approach to efficiently generate more examples could be to use simulated data. However, there is a risk of using simulated data without making sure the data is realistic enough, since a network trained on such data might not perform well when evaluated on real data. The study by Shrivastava et al. [23] mentioned in Section 2.2.4 discussed the importance of using realistic simulated data, and the difficulties in obtaining this. In order to obtain realistic data from simulated data the network GAN was used. The result of of this network was successful, but the method to obtain the images was complex. Though, if generated with high realism Shrivastava et al. showed that training using synthetic data was successful. Hence, it ought to be possible to obtain more data by using simulated data if these are realistic enough.

5.2.4 Future work for Part 2

In summary, the most prominent steps regarding future work for the second part of the project would include:

- Repeated training of each network
- Optimization of training parameters
- Expand the data set

5.3 Ethical and societal considerations

If a future version of the system was to be implemented in a real compartment the ethics regarding the handling of images and the information obtained by the predictions must be considered, since both IR images of the face and entire 3D models of the face would be present. Hence, it is not only a question about handling the IR images correctly, but also the actual depth maps. Depth maps of faces can be used for identification wherefore such information is sensitive. For instance, if the depth maps were of high enough quality it could be possible to detect who the driver is in each car with high accuracy, but how does one ensure that such information is not used in a way that violates personal integrity? On the other hand, using 3D models of the face for identification could be used to open the car or to instantly change the settings of the compartment to suit each individual driver. Moreover, knowledge of who is seated in a certain car can be crucial in case of emergency, for instance ambulance personnel can be more prepared to treat each individual before arriving to a site. Finally, the algorithms presented in this work are not at all excluded to be used in compartments but the same system could be incorporated to produce depth maps in any other area of technology as well. In summary, depth maps can be useful for numerous implementations but for each implementation the aspect of integrity should be considered and all data must be handled carefully.

6
Conclusion

The aim of the project was to investigate and implement algorithms for constructing depth maps using structured light. The depth estimates were compared with a current single point depth estimate, regarding both accuracy and invariance to characteristics of the head. The project also aimed to investigate how depth and pose estimation performed by CNNs were impacted by adding depth maps as input.

From the evaluation of the algorithm in the first part of the project it can be concluded that it was possible to obtain depth estimates with an accuracy of less than one centimeter while keeping high precision, which was aimed for. Moreover, the algorithm seemed to be invariant to characteristics of the head. Indeed, it would be valuable to extend the evaluation process to include a larger number of persons and scenarios.

A comparison was made between a FAb-Net implemented to use IR images as input as compared to a FAb-Net implemented to include both IR images and depth maps as input by early fusion. Both a singletask and a multitask network was implemented. When adding depth maps as additional input to a singletask network for prediction of either depth or rotation the performance was increased, as compared to only using IR images as input. Moreover, adding depth maps as input to a multitask network improved the prediction of facial landmarks, xy-position and rotation. No significant difference in performance could be concluded for prediction of depth.

Bibliography

[1] VTI. "Trötthet hos förare". [Online] . Available: https://www.vti.se/sv/Forskningsomraden/Trotthet. Accessed: 2019-05-03.

[2] W. B. Horng, C. Y. Chen, Yi Chang, and Chun-Hai Fan. "Driver fatigue detection based on eye tracking and dynamic template matching". *IEEE International Conference on Networking, Sensing and Control*, 2004. doi: 10.1109/icnsc.2004.1297400. [Online]. Available: https://ieeexplore.ieee.org/document/1297400. Accessed: 2019-02-28.

[3] Smart Eye. [Online] . Available: https://smarteye.se/. Accessed: 2019-05-03.

[4] P. Zanuttigh, C. D. Mutto, L. Minto, G. Marin, F. Dominio, and G. M. Cortelazzo. *Time-of-flight and structured light depth cameras: Technology and applications*. Springer Publishing, 2016. [Online]. Available: https://www.springer.com/gp/book/9783319309712. Accessed: 2019-02-12.

[5] Kanade T. Lucas Bruce D. "An iterative image registration technique with an application to stereo vision". 1981. [Online]. Available: https://pdfs.semanticscholar.org/51fe/a461cf3724123c888cb9184474e176c12e61.pdf. Accessed: 2019-03-26.

[6] M. Bertozzi and A. Broggi. "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection". *IEEE Transactions on Image Processing*, (no: 1), 1998. doi: 10.1109/83.650851. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/650851. Accessed: 2019-03-26.

[7] S. Sengupta, E. Greveson, A. Shahrokni, and P. H.S. Torr. "Urban 3D semantic modelling using stereo vision". *Proceedings - IEEE International Conference on Robotics and Automation*, 2013. doi:10.1109/ICRA.2013.6630632. [Online]. Available: http://www.robots.ox.ac.uk/~phst/Papers/2013/ICRA2013_1807.pdf. Accessed: 2019-03-01.

[8] The University of Edinburgh - and School of Physics Astronomy. "Theory of Image Processing - Topic 10 : Stereo Imaging". 2006. [Online]. Available: http://www2.ph.ed.ac.uk/~wjh/teaching/dia/documents/stereo.pdf. Accessed: 2019-02-19.

[9] J. Geng. "Structured-light 3D surface imaging: a tutorial". *Advances in Optics and Photonics*, vol: 3. no: 2, 2011. doi:10.1364/aop.3.000128. [Online]. Available: https://www.osapublishing.org/aop/abstract.cfm?uri=aop-3-2-128. Accessed: 2019-02-22.

[10] C Dal Mutto and P Zanuttigh. *"Time-of-Flight Cameras and Microsoft Kinect TM"*. Springer Publishing, 2013. [Online]. Available: http://lttm.dei.unipd.it/nuovo/Papers/ToF-Kinect-book.pdf. Accessed: 2019-02-18.

[11] A. Shamata and T. Thompson. "Using structured light three-dimensional surface scanning on living individuals: Key considerations and best practice for forensic medicine". *Journal of Forensic and Legal Medicine*, vo: 55, 2018. doi:10.1016/j.jflm.2018.02.017. [Online], Available: https://www.ncbi.nlm.nih.gov/pubmed/29471248. Accessed: 2019-02-22.

[12] L. Song, X. Li, Y. G. Yang, X. Zhu, Q. Guo, and . Liu. "Structured-light based 3D reconstruction system for cultural relic packaging". *Sensors (Switzerland)*, vo: 18. no: 9, 2018. doi: 10.3390/s18092981. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/30200665. Accessed: 2019-02-22.

[13] A. Kadambi, A. Bhandari, and R. Raskar. "Computer Vision and Machine Learning with RGB-D Sensors". Springer Publishing, 2014. [Online]. Available: https://www.springer.com/gp/book/9783319086507 Accessed: 2019-02-05.

[14] A. Saxena, S. H. Chung, and A. Y. Ng. "3-D depth reconstruction from a single still image". *International Journal of Computer Vision*, vo: 76. no: 1, 2008. [Online]. Available: https://link.springer.com/article/10.1007/s11263-007-0071-y. Accessed: 2019-03-11.

[15] E. Murphy-Chutorian and M. M. Trivedi. "Head pose estimation in computer vision: A survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vo: 31. no: 4:607–626, 2009. doi: 10.1109/TPAMI.2008.106 . [Online]. Available: http://cvrr.ucsd.edu/publications/2009/MurphyChutorian_Trivedi_PAMI09.pdf. Acessed: 2019-03-01.

[16] J. B. Kuipers. "Quaternions and rotation sequences". Bulgaria. Coral Press. Sofia 2000. 1999. [Online]. Available: http://emis.ams.org/proceedings/Varna/vol1/GEOM09.pdf. Accessed: 2019-05-03.

[17] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew. "Deep learning for visual understanding: A review". *Neurocomputing*, vo: 187, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215017634. Accessed: 2019-02-18.

[18] J W. Cort and M. Kenji. "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance". *Climate Research*, vo: 30. no: 1, 2005. doi: 10.3354/cr030079. [Online]. Available: https://www.researchgate.net/publication/235710066_Advantages_of_the_Mean_Absolute_Error_MAE_over_the_Root_Mean_Square_Error_RMSE_in_Assessing_Average_Model_Performance. Accessed: 2019-05-01.

[19] D.P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". pages 1–15, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980. Accessed: 2019-05-01.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". 2012. doi: 10.1201/9781420010749. [Online]. Available: https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf. Accessed: 2019-05-10.

[21] K. He, X. Zhang, and and J. Sun. S. Ren. "Deep residual learning for image recognition". *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.

[Online]. Available: https://ieeexplore.ieee.org/document/7780459. Accessed: 2019-05-10.

[22] P. Ramachandran, B. Zoph, and Q.V. Le. "Searching for Activation Functions". 2017. doi:10.1186/1472-6963-9-123. [Online]. Available: https://arxiv.org/pdf/1710.05941.pdf. Accessed: 2019-03-01.

[23] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. "Learning from Simulated and Unsupervised Images through Adversarial Training". 2016. doi:10.1109/CVPR.2017.241. [Online]. Available: https://ieeexplore.ieee.org/iel7/8097368/8099483/08099724.pdf. Accessed: 2019-02-22.

[24] O. Wiles, A.S. Koepke, and A. Zisserman. "Self-supervised learning of a facial attribute embedding from video" in *British Machine Vision Conference*, UK. *British Machine Vision Conference*, 2018. [Online]. Avaible: http://www.robots.ox.ac.uk/~vgg/research/unsup_learn_watch_faces/fabnet.html. Accessed: 2019-05-14.

[25] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. "Real-time human pose recognition in parts from single depth images". *Communications of the ACM*, vol: 56. no: 1. 2013. doi: 10.1145/2398356.2398381. [Online]. Available: https://cacm.acm.org/magazines/2013/1/158776-real-time-human-pose-recognition-in-parts-from-single-depth-images/fulltext. Accessed: 2019-03-26.

[26] Z. Gao J. Zhang C. Tang P. Wang, L. Wanqing and P.O. Ogunbona. "Action Recognition from Depth Maps Using Deep Convolutional Neural Networks". *IEEE Transactions on Human-Machine Systems*, (4), vol. 46. no: 4. 2016. doi: 10.1109/THMS.2015.2504550. [Online]. Available: https://ieeexplore.ieee.org/document/7358110. Accessed: 2019-04-05.

[27] L. Spinello A. Eitel, J. T. Springenberg, M. Riedmiller, and W. Burgard. "Multimodal deep learning for robust RGB-D object recognition". *IEEE International Conference on Intelligent Robots and Systems*, 2015. doi: 10.1109/IROS.2015.7353446. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7353446. Accessed: 2019-03-13.

[28] R. Ranjan, V. Patel, and R. Chellappa. "HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. doi: 10.1109/TPAMI.2017.2781233. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8170321. Accessed: 2019-02-22.

[29] N. Otsu. "A Threshold Selection Method from Gray-Level Histogram". vol. 9. no: 1. 1979. doi: 10.1109/TSMC.1979.4310076. [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/moments/moments.html. Accessed: 2019-05-13.

[30] LearnOpenCV. "OpenCV 2.4.3.7 documentation". [Online]. Available: https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/moments/moments.html. Accessed: 2019-05-13.

A

Appendix 1

A.1 Evaluation of depth estimation

Table A.1 shows the measurements used in the evaluation of the depth estimation where 30 tests are carried out for each person and distance. Two test persons are included, noted as A and B respectively in the table. Cells marked in color marks measures where the calculated distance is not within 1 cm from the true distance. Note that all such results occur for person B at distance 50 cm.

Test nr	50 cm A	50 cm B	60 cm A	60 cm B	70 cm A	70 cm B	75 cm A	75 cm B
1	0.4929	0.5016	0.5992	0.5997	0.7054	0.7032	0.7497	0.7481
2	0.493	0.502	0.5953	0.596	0.702	0.7034	0.7458	0.7506
3	0.491	0.502	0.5993	0.5963	0.7019	0.7024	0.7493	0.7483
4	0.493	0.5091	0.5981	0.5964	0.7075	0.7029	0.7492	0.7476
5	0.4929	0.501	0.5984	0.5964	0.7032	0.7036	0.7485	0.7495
6	0.4929	0.4971	0.5962	0.5997	0.7052	0.7027	0.7497	0.7479
7	0.493	0.513	0.5983	0.5964	0.7052	0.7033	0.7489	0.7501
8	0.4929	0.5092	0.5983	0.5966	0.7056	0.7034	0.7496	0.7497
9	0.4908	0.5083	0.598	0.5966	0.7058	0.7033	0.7466	0.7518
10	0.493	0.5005	0.5952	0.5999	0.7057	0.7035	0.7468	0.7496
11	0.4908	0.508	0.5977	0.5965	0.7019	0.7009	0.7493	0.7483
12	0.493	0.5011	0.5983	0.5971	0.7021	0.7017	0.7492	0.7516
13	0.4909	0.4884	0.5983	0.5999	0.702	0.7032	0.7496	0.7472
14	0.4929	0.4838	0.5956	0.5971	0.7018	0.7014	0.7494	0.7476
15	0.4929	0.5095	0.5985	0.5966	0.7058	0.7034	0.7492	0.7475
16	0.493	0.5113	0.5982	0.5971	0.7056	0.7034	0.7461	0.7474
17	0.493	0.5009	0.5979	0.5999	0.7017	0.6985	0.7521	0.7499
18	0.4929	0.4937	0.5954	0.5959	0.7053	0.7011	0.7495	0.7494
19	0.493	0.493	0.5964	0.5998	0.7057	0.703	0.7493	0.7474
20	0.4928	0.4848	0.5987	0.5962	0.7056	0.7033	0.7497	0.748
21	0.4907	0.5008	0.5988	0.5963	0.7055	0.7035	0.7499	0.7501
22	0.4929	0.4943	0.5985	0.596	0.7019	0.7035	0.7497	0.7481
23	0.493	0.4994	0.5986	0.5996	0.7061	0.7028	0.7526	0.7491
24	0.4929	0.4986	0.5983	0.5963	0.706	0.7017	0.7496	0.7477
25	0.4929	0.4933	0.5978	0.5962	0.7058	0.7031	0.7493	0.7497
26	0.4928	0.4922	0.5976	0.5964	0.7057	0.7031	0.7492	0.7496
27	0.4929	0.4924	0.5985	0.5962	0.7019	0.7019	0.75	0.7489
28	0.4928	0.4987	0.59819	0.5963	0.7058	0.7021	0.7495	0.748
29	0.4929	0.4995	0.59811	0.5966	0.7058	0.7015	0.7493	0.7482
30	0.4929	0.5012	0.5979	0.5963	0.7019	0.7036	0.7497	0.7483

Table A.1: Measurements used in the evaluation process of the algorithm for estimating depth.