Learning Similarity for 3D Reconstruction of Intraoperative Environments with Convolutional Neural Networks

Author:
Anita Rau

Supervisor:
Dr. Danail Stoyanov

A dissertation submitted in partial fulfillment

of the requirements for the degree of

MSc Computational Statistics and Machine Learning

of

University College London,

Department of Computer Science.

September 5, 2017

Abstract

Colorectal cancer is the third most common cancer worldwide and, although the fourth deadliest form of cancer, early diagnosis can increase the survival rate significantly. The standard procedure for colorectal cancer diagnosis is colonoscopy; however, the quality of the screenings highly depends on the proficiency of the operator. Computer-aided systems have the potential to improve the accuracy of screenings by mapping the entire inner surface of the colon during the examination. One approach to this would be to use stereoscopic images.

The performance of traditional stereo algorithms depends on the choice of the hand-crafted cost-function that describes how similar two pixels in a stereo image pair are. Convolutional Neural Networks (CNNs) instead, learn similarity in stereo images directly from data. Despite the progress in the field, accurately finding stereo correspondences in surgical images is still very challenging because of reflective surfaces and textureless regions.

In this work, we propose a dense stereo matching algorithm for low-textured regions like the colon and other intraoperative environments. We train different Siamese CNNs and investigate the impact of the size of the receptive field on the accuracy. We show that increasing the receptive field with pooling-layers that condense information from neighbouring pixels enables the model to incorporate more context in low textured regions. Our resulting model yields a more accurate and smooth 3D reconstruction of the colon than previously suggested architectures with small receptive fields, providing better diagnostic information in clinical settings.

Acknowledgements

Special thanks are owed to Patrick for his endless patience, support and optimism, to Evans for his open ears and encouraging words, to George for his impressive English skills, to Mirek for always lifting my mood, to Claudia for simply everything, to Francisco for his brilliant ideas, to Francois for being a friend, and to Dan, for making me part of this wonderful group.

Contents

List of Figures

List of Tables

List of Notations

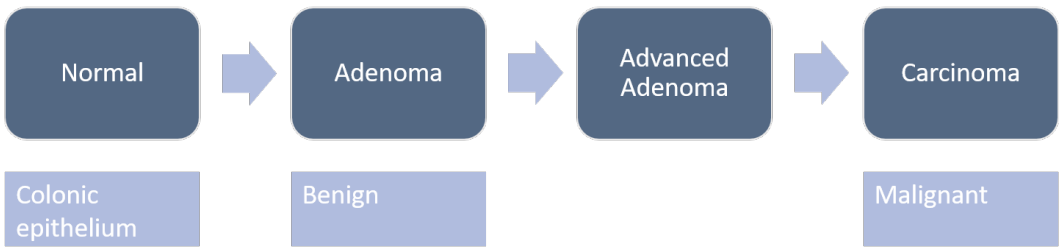
B	Baseline, i.e. the distance between two cameras
$c(\cdot)$	Cost function describing the cost of matching two pixels. Complementary to the similarity s. The higher the similarity, the lower the matching cost
$\mathscr{C}_{comp}/\mathscr{C}_{ref}$	Point cloud $\mathscr{C}_{comp}$ that is compared to a reference cloud $\mathscr{C}_{ref}$
d	Disparity, also used as $d_{i,j}$ to describe the disparity (predicted or known) of a pixel (i,j)
$\mathscr{D}$	Set of all possible disparities
δ	offset parameter of a camera
e	Epipolar line
ε	Reconstruction error
$f(\cdot)$	Function describing the future representation of a pixel. Especially used as output of a Neural Net, or a layer of a Neural Net, which describes nothing else than a feature representation of some input
F	Fundamental matrix
γ	skew parameter of a camera
h	Hidden layer in a Neural Net
$I^L_{i,j}/I^R_{i,j}$	Intensity of pixel (i,j) in the left (L) or right (R) image
$imSize$	Image size
$\mathbb{1}_{\{\cdot\}}$	Indicator function, equals one if the statement in $\{\cdot\}$ is true.
Λ	Intrinsic matrix
$maxDisp$	Highest possible disparity
Ω	Extrinsic matrix
$p_{i,j}$	Pixel at position (i,j) in an image. Also used for 3D coordinates of points in a point cloud
ϕ	Focal length of a camera
Φ	Homography
$s(\cdot)$	Function describing a similarity measure between the feature representations of two pixels
t	Threshold
$Xent$	Cross-entropy $Xent(t,y) = -\sum_{i \in classes} t_i \log y_i$ of predictions y and ground truth t
z	Depth

Chapter 1

Introduction

Computer-aided diagnosis (CAD) is a major research field in medical imaging. The ability to examine the interior of the human body without the requirement of incisions is indispensable for the diagnosis and treatment of many common diseases. CAD has for instance become routine in the detection of breast cancer, it has improved detection of lung nodules, osteoporosis and aneurysms, and has the potential to improve diagnosis in numerous other applications [1]. One of these possible applications is colorectal cancer (CRC) diagnosis. CRC is the third most common cancer in developed countries and accounts for half of all cancer deaths [2]. To improve CRC screening research so far has focused on automatic detection of polyps during colonoscopy using medical imaging techniques [3]. In this thesis we investigate the potential of the 3D reconstruction of the colon as additional tool for CAD in CRC diagnosis.

In this introductory chapter we first summarise colorectal cancer, we explain current standard screening methods and discuss why a 3D model of the colon can be a valuable addition to those. We then give an overview over current advances in the 3D reconstruction of all types of *in vivo* scenes and the colon in particular. Finally, we summarise the objectives of this thesis and give an overview over the structure of this work.



Different studies estimate this process to take about 10 - 15 years. The transition between the stages is triggered by alterations like mutations in the DNA. Around 30% of the population over 60 years has adenoma [5].

Figure 1.1: Pathway from healthy colonic epithelium to carcinoma adapted from [6].

1.1 Introduction to Colorectal Cancer

Colorectal cancer (CRC) describes the uncontrolled growth of cells within the colon or rectum [4]. The development from a trigger in a single cell to a tumour occurs stepwise and is believed to take 10-15 years [5]. A scheme of the process that leads from healthy mucosa to cancerous cells is shown in figure 1.1. During this process healthy mucosa first develops to benign adenomas, referred to as polyps [6], and subsequently to carcinoma; however, not all adenomas have to become cancerous. An example of the development is shown in figure 1.2. It depicts the same polyp observed before and after a period of 3 years, during which the diameter of the adenoma increased from 5 mm to 7 mm.

The causes for CRC are not clear but a number of risk factors have been identified, amongst them are nutrition, age, and genetic factors [5]. Common symptoms of CRC are change in bowel habit or rectal bleeding. However, due to their similarity to common colorectal complaints, the symptoms of CRC do not indicate an unambiguous diagnosis [7]. In fact, most people who suffer from, for example, a change in bowel habit, do not have cancer [5]. The ambiguotiy of the symptoms can lead to a severe time lag between the outbreak of the disease and its treatment. This is especially dangerous, as progressed CRC was found to have a 5 year survival rate of only 8.1%, whilst earliest stages of CRC were estimated to have a survival rate

Images of a colorectal adenoma increasing in diameter from 5mm (left) to 7mm (right) within 3 years. Each stripe of the measure corresponds to 1mm.

Figure 1.2: Polyp increasing in diameter [6].

of 90% [8]. The lack of symptoms together with the necessity of early diagnosis therefore suggest the importance of CRC screening.

1.2 Colorectal Cancer Screening

The standard procedure for CRC screening is colonoscopy. During colonoscopy the inner surface of the colon is examined with a colonoscope. In figure 1.3 the

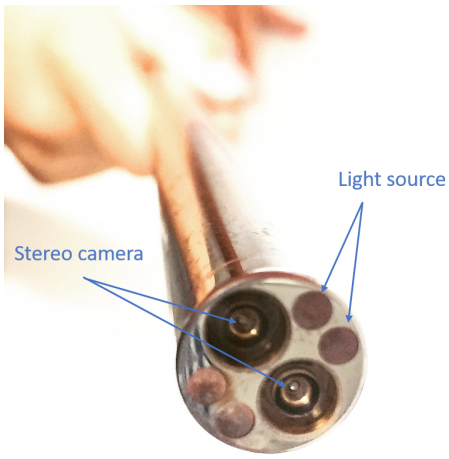


Figure 1.3: Stereo endoscope of the DaVinci Surgical System.

endoscope used on a phantom of a colon in the present project is shown. It has a stereo camera and a light source that is divided into four light points at its end that enable capturing a video sequence of images. It is part of the DaVinci Surgical

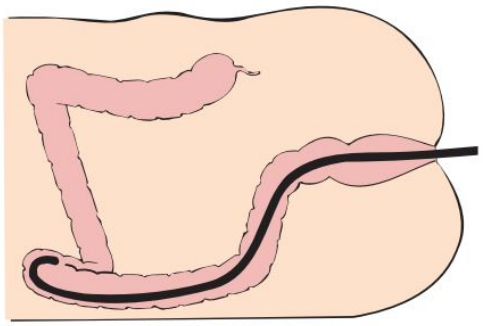
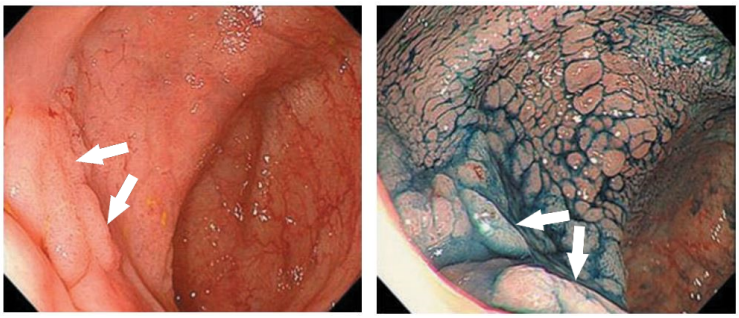


Figure 1.4: View of the endoscope inserted into the colon during colonoscopy [6].

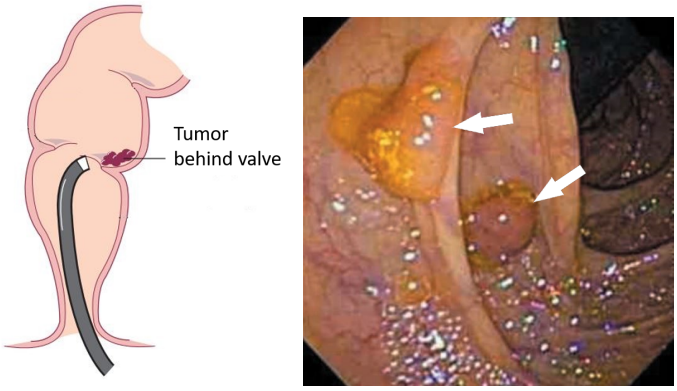
System, a robot for non-invasive surgery. For *in vivo* colonoscopy an endoscope has to be flexible and able to follow the shape of the colon as shown in figure 1.4. The images that are captured with the endoscope are displayed on a monitor and can be analysed in real-time. An experienced endoscopist can identify a majority of all polyps in the images without further post-processing and assess the necessity of their removal.

However, the detection of flat or occluded polyps in the colon remains challenging for reasons that are depicted in figure 1.5. In image 1.5 (a) one can observe that, for example, flat adenomas are alomost impossible to locate with an untrained eye. Only after the use of indigo carmine the boundaries of the lesion in the left lower corner become visible. Image 1.5 (b) shows another difficulty that is characteristic for colonoscopy. Its shape occludes surfaces located behind folds and valves for a forward pointed endoscope. Although the endoscope can be oriented to observe these areas as shown in image 1.5 (c), the examination of reverse surfaces is not assured.

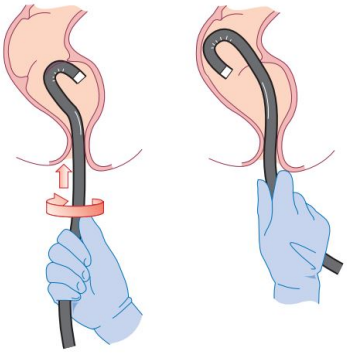
These challenges are the reason why the quality of colonoscopy highly depends on the proficiency of the operator. A study from 2010 showed that the risk of developing colorectal cancer within 5 years after colonoscopy was significantly higher among subjects who were treated by endoscopists with a low adenoma detection rate. This rate is the proportion of screened subjects in whom at least one adenomatous lesion was identified [9]. In a different study Pickhardt *et al.* found that 10.8% of polyps with a size of at least 5 mm were missed by experienced operators [10]. A



(a)



(b)



(c)

(a) Left: Ordinary colonoscopic view. Right: Chromoendoscopy with indigo carmine. (b) Left: Tumor occluded by a valve. Right: Typical appearance of the right colon during retroflexion, demonstrating two polyps on the proximal side of a fold. Only the tip of one polyp had been visible on forward view. (c) Technique for retroflexion.

Figure 1.5: Challenges of colonoscopy. Images adapted from [6].

groundtruth was obtained by performing a CT colonography (CTC), and validating the indicated polyps by re-examining the colon during a second colonoscopy. CTC is a computed tomography based screening test for colorectal cancer, recently reported to detect significantly less polyps than colonoscopy [11]. In a similar study [12] a 4% missing rate for lesions of at least 10mm was reported. As the endoscopists were aware of the assessment in both studies, it can be assumed that their performance is worse under normal conditions, resulting in even higher missing rates.

Consequently, a decrease in the missing-rate of polyps during colonoscopy is crucial for the quality of early diagnosis, and thus for the improvement of the treatment and the survival rate of colorectal cancer. To improve the quality of colonoscopy, computer aided methods have been researched. Although it is not current practice as yet, to post-process images captured during colonoscopy, an analysis of the images have shown great potential. Segmentation algorithms are able to detect up to 100% of all polyps on observed surfaces [3].

But even if all polyps on observed surfaces could be detected, unintentionally unexamined surfaces remain a risk. 3D reconstruction of the colon has the potential to minimise this risk. Obtaining a 3D model during colonoscopy can reveal unexamined surfaces during the procedure and provide instructions to the operator in real-time guiding the colonoscopy in a manner that guarantees the observation of the entire inner surface of the colon. In Pickhardt's study mentioned above, they not only assessed the missing rate of polyps but also analysed the location of missed adenomas and found that the majority was located on the proximal, the less visible side, of folds or in the rectum [10]. The 3D reconstruction of the colon could therefore improve the detection rate of rear polyps considerably.

Moreover, a 3D model of the colon allows an exact localisation of polpys within it. This can assist surgeons in visualization and pre-op surgical planning [13] and could be an important step towards automated robotic interventions. Better 3D guidance of the lesion removal can further reduce complications, like perforation of the colon, during polypectomy [14].

1.3 A Brief Review of 3D Reconstruction of *In Vivo* Scenes

The reconstruction of *in vivo* surfaces has been researched especially in the context of computer-assisted laparoscopic surgery. Laparoscopy allows to examine the abdomen through small incisions which reduces surgical trauma and hospitalization as opposed to open surgery. Because the procedure faces similar challenges as colonoscopy - mainly complicated scenes, and lack of fronto-parallel surfaces with unique colours and texture [15] - advances in this field are a useful starting point for the reconstruction of the colon.

Obtaining the shape and morphology of soft-tissue surfaces *in vivo* has proved to be a challenging task. The most established approach for recovering the shape of an object is stereo reconstruction. Stereoscopy was shown to outperform other techniques regarding point density and conservation of fine details in laparoscopic scenes. Stereo methods can be broken down into the following steps [16]:

- camera calibration
- acquisition of stereo images pairs of the scene
- establishing stereo correspondences of the points in the images
- triangulation of the pairs using the known properties of the cameras

While steps one, two, and four are widely solved problems, step three remains challenging. Instead of establishing correspondence for each pixel pair independently by minimising a cost function, it has been shown that including biologically inspired cost aggregation over a window of pixels improves the reconstruction [17]. Recent stereo algorithms, designed especially for laparoscopy, extend the window approach and take global smoothness constraints into account when establishing the stereo correspondences [18, 15, 19]. They all follow the idea that neighbouring pixels are likely to be the projection of the same object with consistent depth. While some approaches are based on finding initial, easy matches and propagating their

depth to neighbouring pixels in a iterative matter, others match all pixels initially and smooth the results afterwards.

Although stereo algorithms outperform other techniques on *in vivo* scenes, namely structured light or time-of-flight [16], to the best of our knowledge they have not yet been applied to the reconstruction of the colon. More recent research has mainly focused on reconstruction from frames obtained from a video capsule that is swallowed and traverses the colon [20, 21, 22]. However, this approach suffers from a low frame rate of 2 - 6 frames per second and uncontrolled, unrestricted motion. Other approaches evaluate monocular endoscope images [23, 24], or incorporate knowledge about the geometry of the folds typically appearing in the colon [25].

1.4 Dissertation Objectives

In this dissertation we provide a stereo algorithm for the 3D reconstruction of the colon, to obtain improved diagnostic information and enable better treatment of colorectal cancers. We focus mainly on improving the initial stereo correspondences in particular by increasing the accuracy and density of the matches between the image pairs. We are the first ones to apply a stereo algorithm to the reconstruction of the colon. The main objective of this thesis is to exploit recent advances in Deep Learning and to learn a similarity measure that establishes stereo correspondences directly from data.

As scenes from colonoscopy lack texture and edges we will introduce a novel Neural Network architecture that enables the model to incorporate more context while preserving the resolution of the images. We study the impact of different network architectures on the accuracy of our 3D reconstruction. We investigate how the depth, the number of pooling layers, and implicitly the size of the receptive field of the network influence our results. In particular, we will examine how several architectures perform on different types of scenes regarding density and preservation of detail in the reconstructed model. We further introduce a way to incorporate the certainty about predicted matches and experiment with different lower thresh-

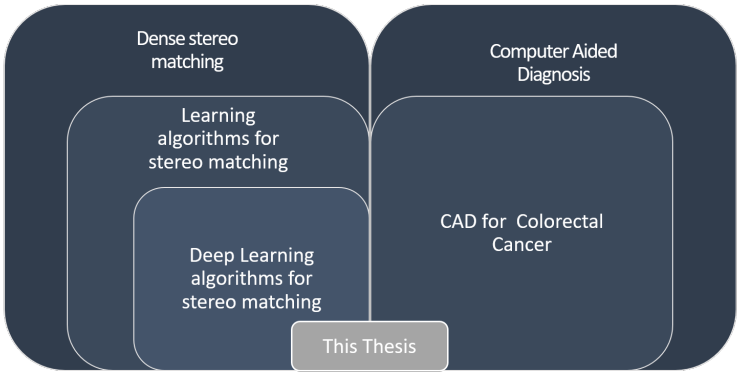


Figure 1.6: Thesis structure

olds to regulate noise in our reconstructions. We will introduce different evaluation matrices and evaluate our findings on stereo datasets of a colon phantom and real intraoperative scenes.

Figure 1.6 puts our contribution into context and gives an overview over the structure of this thesis. In chapter 2 we introduce geometric properties of stereo vision and triangulation. We review standard approaches to the problem and find that hand-crafted solutions could potentially be outperformed by Convolutional Neural Networks. In chapter 3 we introduce Deep Learning techniques and review current approaches. In chapter 4 we introduce our new approach to the reconstruction of the colon. The results of our experiments are discussed in chapter 5. Lastly, we summarise our findings in chapter 6 and give an outlook to future work.

The code used for this project can be found here: `https://github.com/anitarau/Stereo_matching`

Chapter 2

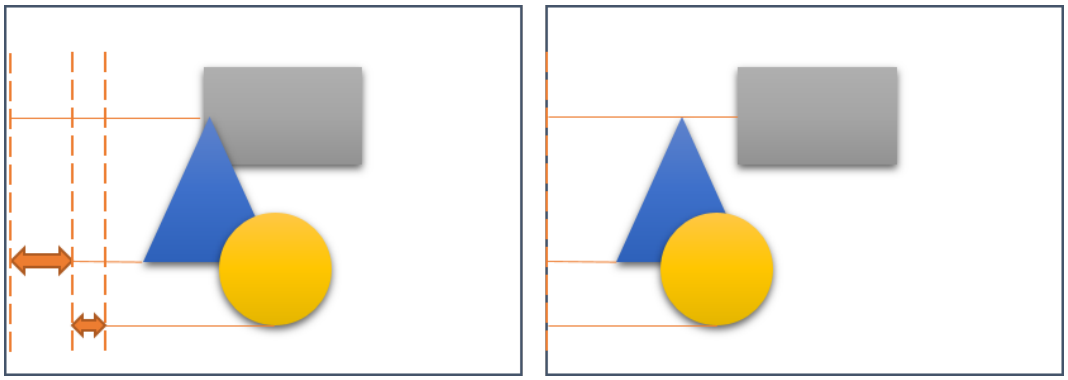
Geometric Properties of Stereo Vision

In computer vision we try to extract information from images. One much researched and still unsolved application is the reconstruction of a 3D scene. There are different approaches to obtain a 3D model from a scene, for example inference from a single image [26], from structured light [27] or from time of flight [28]. In this thesis, we focus on obtaining a 3D model from a pair of images captured with a stereo camera. In this chapter we discuss basic concepts of stereo vision. We start by giving an intuition of how depth can be inferred from stereo images. We will see that inference follows the same principles as human stereoscopic vision. We then look at the geometrical description of binocular vision and lay the groundwork for stereo matching algorithms, that aim to match each point in the left image with a point in the right image of a stereo image pair. In the second part of this chapter, we then discuss which approaches have been proposed to tackle the problem of 3D reconstruction from stereo image pairs.

2.1 Disparity and Depth

Similar to human vision, one can infer distances in a scene from a pair of images taken with a stereo camera. Closing one eye at a time and focussing on a nearby object one can perceive the object *jumping* horizontally from one side of the field of view to the other when switching eyes. Focussing on a distant object, the distance

(a) Left and right view on three objects. Our brain infers that the relative depth between the circle and the triangle must be smaller than the relative depth between the triangle and the rectangle.



(b) The distances from the left edge of the left image to the dashed lines represent the disparities of the objects. The orange arrows indicate the relative disparities. One can clearly observe that the relative disparity between the circle and the triangle is smaller than that between the triangle and the rectangle. As a consequence the relative depth between the circle and the triangle appears smaller.

Figure 2.1: Stereo view on three objects in the space

the object moves between the two views is considerably smaller. This distance is referred to as disparity in an image pair. The human brain is capable of measuring relative disparities between objects and estimating depth from these. We intuitively know, that objects with greater disparity must be closer. Similarly, if the disparities are known in a pair of images taken by a stereo camera, the geometry of the scene can be inferred. But in order to obtain disparities, one needs to match every pixel in the left image of a stereo pair to the corresponding pixels in the right image. This task will be referred to as the correspondence or matching problem for the remainder of this work [29, 30].

If the correspondence is known and the disparity d obtained, then the depth

can be derived by means of equivalent triangles [31]. Let B denote the baseline, that is the distance between two cameras, and let ϕ denote the focal length, that is the distance from the optical centre to the image plane. Then the depth z of the corresponding point in the world can be described as

$$z = \frac{\phi B}{d} \tag{2.1}$$

2.2 Epipolar Constraint

Geometrically, the relation between two images of the same scene is described by the *epipolar constraint*. In figure 2.2 a scene w is projected into two offset cameras. The left camera projects w onto x_1, the point where a ray through w and the optical center 1 intersects the camera plane. Because they all lie on the same ray, each of the four depicted world states w are projected onto the same point x_1. Consequently, it is not possible to infer the distance of the object by means of a single camera view. However, each w is projected onto a different point on the camera plane of the right camera. An object w can therefore be located in a 3D environment as the intersection of the ray that passes x_1 and optical center 1, and the ray that passes x_2 and optical center 2. As long as the two cameras are not translated purely along or perpendicular to their optical axes, the optical centre of camera 1 must intersect each ray through the camera plane of camera 2. As a result, inference of the depth is possible. To find x_2 on the right camera plane we observe that all world states w lie on the epipolar line in image 2. As a result, we only have to look for x_2 along one dimension [30]. We exploit this property further by introducing the concept of rectified images. The idea is to modify the images in a way, that allows the epipolar lines to lie on a horizontal line. For the rectification we need a *homography* Φ_2, a transformation describing the relation between two images, that maps the epipole e_2 to $[1,0,0]^T$. Applying this transform maps each point x_{i2} in the right image to its rectified version

$$x'_{i2} = \hom(x_{i2}, \Phi_2). \tag{2.2}$$

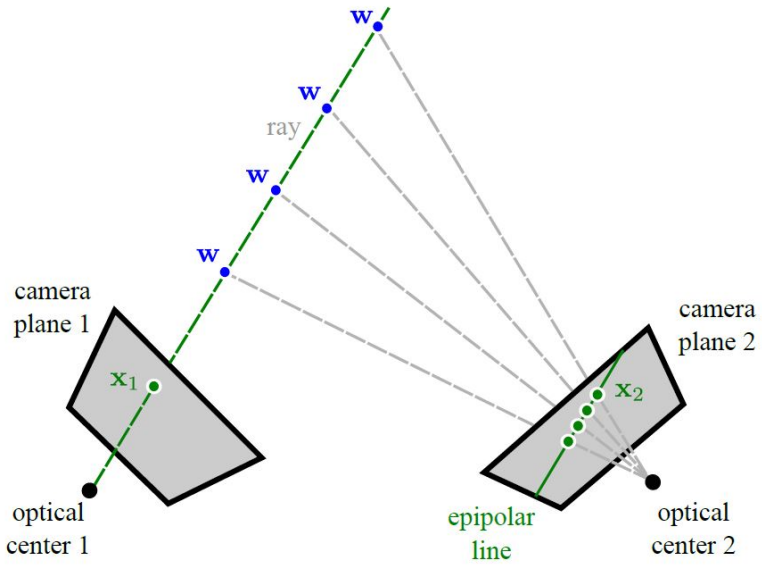


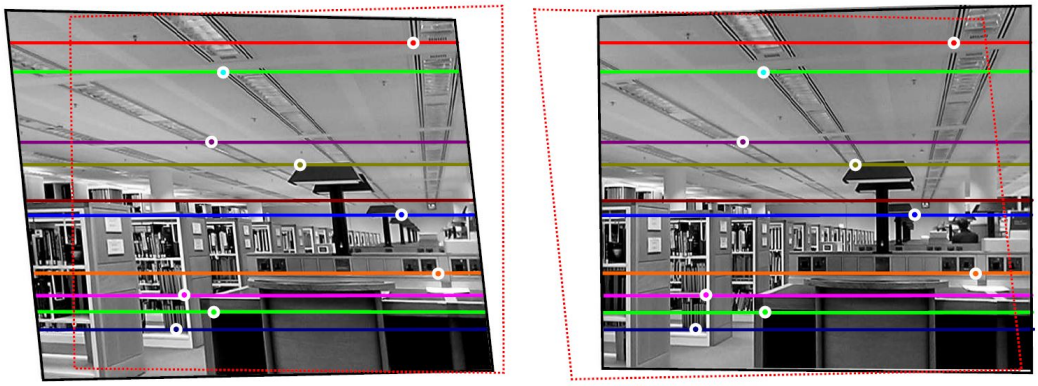
Figure 2.2: Epipolar line in stereo view [30]

Rectifying the left image requires the computation of the Fundamental matrix F, a linear transformation that puts the points in the left and right image into relation. The matrix F satisfies the equation

$$\tilde{x}_2^T F \tilde{x}_1 = 0 \tag{2.3}$$

where $\tilde{x}_1$ and $\tilde{x}_2$ are the homogeneous coordinates of a point in image 1 and its counterpart in image 2. Then a second homography Φ_1 needs to be obtained that aligns the epipolar lines in image 1 with those in image 2. Details of the intermediate steps can be found in chapter 16.5 of [30]. The result of the rectification is shown in figure 2.3.

The result of this process is the useful property that the search space of the corresponding point x_2 in the right image to the point x_1 in the left image is reduced to a horizontal line through x_1. Many stereo matching algorithms take advantage of this property as we will see in the following sections. For the remainder of this dissertation images are assumed to be rectified.



Each point in the left images induces a horizontal epipolar line in the right image and vice versa. The dotted line is the superimposed outline of the other image.

Figure 2.3: Rectified stereo image pair [30]

2.3 Triangulation

To derive 3D coordinates from images we first need to know the relation between world coordinates and camera coordinates. For a single camera this relation can be described as a linear projection equation if both, the image and world coordinates, are given as homogeneous coordinates. Let $[u,v,w]$ be the coordinates of a point in the world and let $[x_j,y_j]$ be the 2D coordinates of the projection of the same point onto the image plane of camera j. Then a camera j satisfies the *pinhole camera model* [30]

$$\lambda \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \phi_{xj} & \gamma_j & \delta_{xj} \\ 0 & \phi_{yj} & \delta_{yj} \\ 0 & 0 & 1 \end{bmatrix}}_{\Lambda_j} \underbrace{\begin{bmatrix} \omega_{11j} & \omega_{12j} & \omega_{13j} & \tau_{xj} \\ \omega_{21j} & \omega_{22j} & \omega_{23j} & \tau_{yj} \\ \omega_{31j} & \omega_{32j} & \omega_{33j} & \tau_{zj} \end{bmatrix}}_{\Omega_j} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \tag{2.4}$$

where Λ_j denotes the *intrinsic matrix* and Ω_j the *extrinsic matrix* of camera j. The intrinsic matrix describes properties of the camera and can be obtained through calibration [32]. The focal length ϕ describes the distance of the optical centre to the image plane. The offset and skew parameters, δ and γ, account for an image plane which is not perfectly perpendicular and centred. The extrinsic matrix describes the exterior orientation and can be learned by plugging in known pairs of image and

world coordinates, and learning the solution to the algebraic problem. This can be used as starting point for the non-linear optimisation of the geometric problem.

If the intrinsic and extrinsic matrices of both cameras are known then a new world point $[u, v, w]$ can be inferred. This is known as *triangulation*. Solving the last line in equation 2.4 for λ and plugging in the expression for λ we get a linear system with two equations and the three unknowns u, v, w. Using two cameras and corresponding image points we get four equations. This system can then be solved and optimized non-linearly to obtain the most likely world coordinates $[u, v, w]$.

The challenge of obtaining the corresponding image points $[x_{i1}, y_{i1}]$ and $[x_{i2}, y_{i2}]$ of camera 1 and 2 for a 3D point i is the focus of our work. We introduce approaches to this problem in the following section, before we introduce Deep Learning and our approach to the corresponding problem in the following chapters.

2.4 Review of Methods to Establish Correspondence

Establishing correspondence in stereo images is non-trivial due to a number of factors, for instance

- occlusions due to the geometry of the scene
- specular highlights due to the lightening conditions
- repetitive patterns
- the lack of texture and edges

To classify different approaches to the correspondence problem, Scharstein *et al.* presented a taxonomy in 2002 identifying four main sub tasks for deriving depth maps from stereo image pairs [33]. Roughly, the intermediate steps were

- (1) computation of matching costs
- (2) cost aggregation
- (3) disparity selection
- (4) disparity refinement

Together, these four steps establish stereo correspondences between points in images. There are many efficient approaches to steps three and four, namely to optimally select the disparities once the matching costs in images pairs are found. Common solutions are graph cuts [34, 35, 36] or belief propagation [37], which have been researched beyond the stereo matching application. More recently research in the field of stereo vision for 3D reconstruction has focused on step one.

In the following, early approaches to the stereo matching problem are discussed. We will first consider sparse matching algorithms, which are computationally easy but insufficient for 3D reconstruction. We then take a look at earlier learning algorithms and discuss how they improved results. In chapter 3 we eventually analyse how recent Deep Learning approaches yield state of the art results.

2.4.1 Early Approaches to Stereo Matching

(1) computation of matching costs

Early work on stereo matching included a pre-processing step in which feature points were extracted from the pair of images that were easily matchable, like edges and corners. This sparse approach to the matching problem was computationally easier but required yet another intermediate step namely the interpolation between feature points to match all pixels of the image [38]. The approach was to extract interest points in both images based on intensity discontinuities that are likely to occur at points of depth discontinuity. In a second step those interest points which appear in both images of the stereo pair are identified and matched [39].

Hsieh *et al.* propose for example a 1D matching problem. They extracted gradient and intensity signals along each epipolar line and matched peaks and valley along them. Figure 2.4 gives an intuition how gradients and intensities can be used to find points of interest that can potentially be matched.

For the 3D reconstruction of a scene with high variance a sparse representation is not sufficient. We thus try to find the corresponding pixel in the candidate image to each pixel in the reference image instead of focussing on interest points, like edges [40]. This approach is called dense stereo matching.

To determine how likely a pixel corresponds to one of the candidate pixels, a

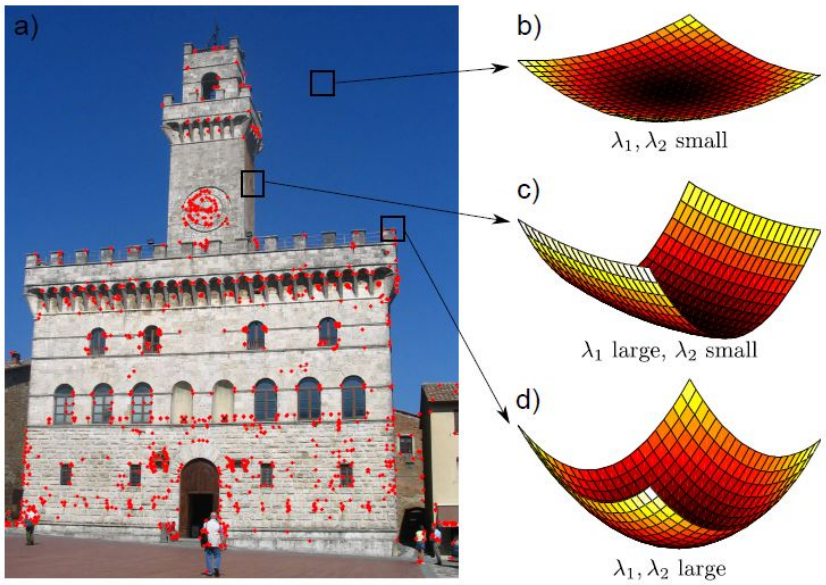


Figure 2.4: Gradients around interest points can reveal context.

Around corners (d) the gradients in two orthogonal directions are large. Around edges (c) the gradient along one direction is large. Around textureless points (b) the gradients are small. Image (a) depicts all edges detected with a Harris corner detector. Details in [30].

similarity measure needs to be selected [40]. Geometrically, this correspondence is established if the pixels in both patches are projections of the same 3D scene [41]. Algebraically, the correspondence is described by means of hand-crafted matching cost functions. Different algorithms have been proposed that compare a pixel in the reference image (usually left) to all possible corresponding pixels in the candidate (right) image finding the match with the lowest cost, or equivalently, with the highest similarity score. Formally, the similarity measure can locally be descried as a function

$$s(f(I^L_{i,j}), f(I^R_{i,j-d})) \tag{2.5}$$

where $I^L_{i,j}$ and $I^R_{i,j-d}$ denote the intensity of the pixel $p_{i,j}$ in the left image, or the same pixel shifted by a disparity d in the right image, respectively. The function $f(\cdot)$ is some feature representation of the intensities of the pixels or a window around them. Figure 2.5 illustrates the similarity measure $s(\cdot)$. The true match of the highlighted pixel in the left image is the dark blue pixel in the right image. The

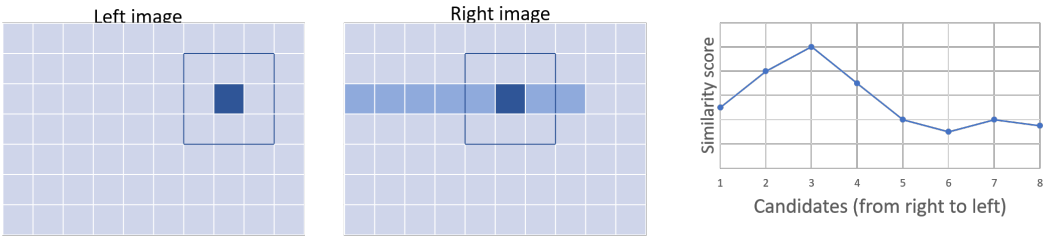


Figure 2.5: Matching with similarity measures.

other highlighted pixels in the right image are potential candiates. The graph on the right hand side depicts some similarity function over a window around the pixels of interest. Ideally the maximum similarity score corresponds to the true match, here the third pixel from the right. Let for simplicity of notation $\mathbf{x} := f(I^L_{i,j})$ and $\mathbf{y} := f(I^R_{i,j-d})$. Some naive cost function are:

Sum of squared intensity differences [41]: This approach minimises the squared error between a target pixel and the candidate pixels. The similarity is highest when the squared error is lowest. The cost function is defined as

$$c(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||_2 \tag{2.6}$$

Binary matching costs [29]:

$$c(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \text{if} \quad \mathbf{x}=\mathbf{y} \\ 1 & \text{otherwise} \end{cases} \tag{2.7}$$

Normalized cross-correlation [42]:

$$c(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{||\mathbf{x}||\cdot||\mathbf{y}||} \tag{2.8}$$

Comparison of gradient fields [43]: Let $\mathbf{x}$ and $\mathbf{y}$ be two gradients at a certain location in the left and right image. Then the average magnitude that indicates confidence is defined as $\bar{\mathbf{m}} = \frac{(|\mathbf{x}+\mathbf{y}|)}{2}$ and similarity is represented as negative

difference $-\mathbf{d} = -|\mathbf{x} - \mathbf{y}|$. The evidence for a match is then given as

$$s(\mathbf{x}, \mathbf{y}) = \bar{\mathbf{m}} - \mathbf{d}. \tag{2.9}$$

Non-parametric measures [44]: For example, the rank transform which is defined as the number of pixels within a local region whose intensity is less than the intensity of the centre pixel.

The selection of the cost function is crucial for optimising the matching in different conditions. While gradient-based functions can deal with images that have both, areas of high and low texture, non-parametric approaches can more easily deal with outliers and are robust towards changes in the properties of the hardware [33].

(2) cost aggregation

The second intermediate step of stereo matching algorithms in Scharstein's taxonomy, cost aggregation, only applies to methods that use a local approach to the post-processing described in the following section. Cost aggregation refers to the spatial aggregation of initial matching costs over a support region around the pixel of interest. A standard way to perform cost aggregation is to replace the matching cost of each pixel p with the average cost of the pixels in a window around p for a given disparity [45]. In areas without texture, like a white wall in the background, it is hard to infer matching points, because pixel-wise cost calculation is ambiguous. Cost aggregation therefore aims to incorporate information from surrounding pixels within a certain window. However, standard approaches assume constant disparities within this window, and summing or averaging over a region effectively blurs the image such that information along edges or corners is lost [46]. To balance the trade-off between aggregating information in areas of low texture and blurring edges, there are several approaches to the particular execution of the aggregation, for example shift-able windows [47] or windows with adaptive size [48].

(3) disparity selection

Once the cost-function is computed for all disparities, there are different approaches to define and select the *optimal* disparity. Local approaches evaluate the function values obtained by cost aggregation and choose the disparity with the lowest costs, this is called the *winner-take all* (WTA) optimisation [33]. The aggregation serves as a way to implement smoothness by looking at areas rather than single pixels. However, a major drawback of the WTA approach is that several pixels in the reference image can be mapped to the same point in the candidate image [40].

To extend on the smoothing idea, global optimisation approaches were proposed. Instead of looking at areas separately, all disparities for the entire images are optimised simultaneously and collectively. A common approach are Markov Random Fields (MRFs) [49]. These define the global cost function, or energy, as sum over all unary costs and all pairwise costs. Unary costs describe the cost of a disparity being in a particular state and they are negative proportional to the probability of a disparity given the input data. Pairwise costs incorporate smoothness by means of an appropriate prior [30]. Minimisation methods for this global cost function have been researched thoroughly. Most common algorithms employ graph cuts [50], which is when performed exhaustively NP-hard, and its computationally more feasible approximation alpha-expansion [34].

Szelski *et al.* compared different energy minimization algorithms in [49] and highlighted other notable algorithms such as loopy belief propagation [51] and tree-reweighted message passing [52], with performance depending on the application. However, in a similar work [53] Kolmogorov *et al.* found that graph cuts performs best.

There is a range of algorithms between strictly local or global optimisation, like for example Semi-Global Matching [46]. Hirschmueller *et al.* proposed to calculates matching costs pixel-wise but to employ globally constrained smoothness.

(4) disparity refinement

After the correspondence points are found the resolution of the depth map can be increased. One way to find sub-pixel disparities is to fit a curve to the matching

costs given at discrete disparity values for example with the iterative intensity interpolation algorithm proposed in [54].

2.4.2 Learning the Matching Costs

In the previous section we summarised early approaches to the correspondence problem. In this section we will look at algorithms that have the same goal but use machine learning techniques to leverage training data and learn aspects of the matching cost function, instead of using hand-crafted functions and hand-tuned parameters.

Some of the first work on stereo matching using statistical learning was done by Kanade *et al.* [48] who employed an adaptive window size in the cost aggregation step of the stereo matching algorithm. This extension allowed them to encounter the trade-off between textureless regions and blurred edges. Higher uncertainty about the disparity is assigned to pixels with higher distance from the centre of the window by means of a Gaussian distribution with zero mean, and a variance that is learned from the data. This allows to model variation of disparities within a window and, together with intensity variation, to adapt the window size such that uncertainty is minimised.

Both, Cheng *et al.* [55] and Zhang *et al.* [56] proposed a Bayesian framework to improve performance of MRF algorithms based on graph cuts or belief propagation. Both formulated a maximum a posterior (MAP) problem in which the MRF parameters and the disparity map were learned from a stereo image pair.

Kolmogorov *et al.* [57] use stereo to improve segmentation in video sequences. They incorporate a colour likelihood modelled as a Mixture of Gaussians over RGB values whose parameters are learned from frame to frame in the MRF framework.

Scharstein *et al.* [58], and Li *et al.* [59] were the first ones to treat the correspondence problem as supervised learning task, incorporating ground-truth disparities. Scharstein constructed a dataset with ground-truth disparities and extended the MRF approach by learning the parameters of a discriminative Conditional Random Field (CRF). In particular, the unary costs and the pairwise costs were modelled as a linear combination of feature functions that was learned from the training images.

Li used a structured support vector machine to automatically learn non-parametric cost functions in a CRF to discard the limitations imposed by particular assumptions about the form of the disparity distribution. Expressing the total cost as inner product of a feature vector and a vector of the corresponding costs, the kernel trick was exploited and the necessity of a parametric representation avoided.

Chapter 3

Deep Learning for Stereo Matching Algorithms

”Inventors have long dreamed of creating machines that think. When programmable computers were first conceived, people wondered whether such machines might become intelligent, over a hundred years before one was built” [60]. These are the words with which Goodfellow *et al.* begin their Deep Learning book. The society’s perception of Artificial Intelligence (AI) has since paved the way for a ”thriving field with many practical applications and active research topics”. In the 1960s ”intelligent” systems started their triumph by solving problems that are hard for the human brain to solve. For example IBM’s Deep Blue defeated world champion in chess Garry Kasparov in 1997 [61]. But it turned out that these intellectually challenging task were trivial for computers: ”Ironically, abstract and formal tasks that are among the most difficult mental undertakings for a human being are among the easiest for a computer”.

In an attempt to define Machine Intelligence Legg and Hutter [62] claimed that ”A fundamental problem in artificial intelligence is that nobody really knows what intelligence is”. They suggested to define intelligence as a measure of an ”agents ability to achieve goals in a wide range of environments”. Goodfellow *et al.* elaborated on this idea in the context of AI and found: ”The true challenge to artificial intelligence proved to be solving the tasks that are easy for people to perform but hard for people to describe formally - problems that we solve intuitively,

that feel automatic, like recognizing spoken words or faces in images."

In the light of this understanding of machine intelligence, algorithms have since focussed on learning from experience. Many common algorithms and methods in Machine Learning are based on the idea of learning patterns from large data sets with many examples instead of understanding the underlying mechanism conceptually. Examples are Logistic Regression, Random Forests or Support Vector Machines [63]. But all of these approaches rely on the selection of appropriate feature representations.

Let us assume we want to predict house prices in Boston. To infer those we would likely look at the size of the houses, on their distance to schools and the area where they are located. But how can we classify data whose feature representation is very complex? If we want to classify an object in an image, like a car or a dog, our learning algorithm needs to be able to discard noise like the angle, the posture, the colour or the lightening. But it is not straight forward to extract such high-level features from raw data. This is where Neural Networks had their appearance. They are able to form very complex representations from simple ones [60] and to discover intricate structures in high-dimensional data [64]. When computational power had become strong enough at the beginning of this century, 10^6 times stronger than when they were first mentioned to be precise, deep Neural Nets proved to be very useful [65]. They now form the class of Deep Learning, which today outperforms all other algorithms in many tasks like object recognition, speech recognition, natural language understanding, or image understanding [66, 67, 68, 69, 70, 64].

3.1 Introduction to Neural Networks

Artificial Neural Networks (NNs) obtained their name because they schematically represent neurons in the brain: A signal arrives at a neuron, is processed, and passed to the next neuron if the processed signal exceeds a threshold.

In NNs each neuron (or unit) maps the input of incoming signals to a weighted

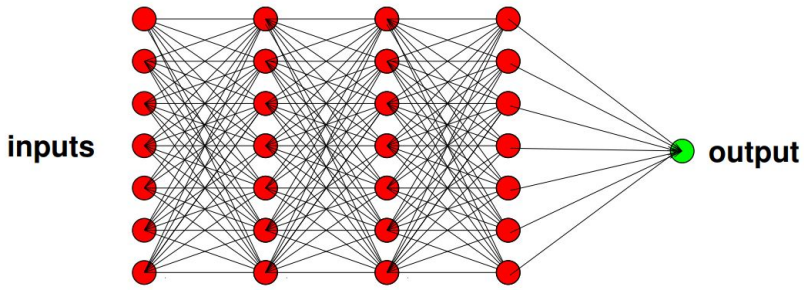


Figure 3.1: Illustration of a Neural Net [71]

combination of those:

$$h_j = f(\sum_i w_{ij} h_i) \tag{3.1}$$

where w_{ij} denote the weights. The output of a node is an input to a node of the subsequent *hidden layer*. Instead of writing $h = f_1(x)$ where x is the input to the network and $y = f_2(h)$ is the output, we could also write $y = f(x)$, where $f = f_1 \circ f_2$. In this manner we can stack many layers, or functions respectively, to get a more complex mapping.

In figure 3.1 we illustrate a network consisting of several layers of units, indicated by red dots. The more layers we *stack*, the *deeper* we call our model. The crucial property of the functions f_i is that they cannot be linear. Otherwise there exists a $\tilde{W}$ such that $\tilde{W}x = W_1 W_2 ... x$ and we can learn the same representation with just one layer. In a standard feed-forward NN usually the functions

$$f_i(x) = \sigma(W \cdot x + b) \tag{3.2}$$

are used (in matrix notation). W and b are referred to as the weights and biases of the network, they are initialized randomly and subsequently learned during the process of training. The function $\sigma(\cdot)$ is the sigmoid function, the *activation function* or *non-linearity* of the network. Other activation functions exist, for example the computationally cheaper rectified-linear unit (ReLU) [64].

Activation functions assumed their name because they push small outputs to

zero and large outputs to 1. The result is, that only signals that are strong enough are activated and forwarded. Cybenko showed that a one layer Neural Net with enough units can approximate every possible function [72]. Intuitively, we add up enough different shifted and scaled sigmoid functions to approximate any function. The goal (of a standard NN) is to learn a function $f = f_1 \circ f_2 \circ ...$ that maps any input into its correct class.

In general there are two options to improve the vanilla network from figure 3.1: widening the network or deepening it. We have just seen that theoretically one layer is enough to approximate f, however given limited computational power, a deep network can be a lot more powerful. It allows to break down a complex functional mapping into series of smaller and easier computations, and could therefore learn a much better representation with the same computational costs. In practice usually a compromise can be found, balancing width and depth.

As in most supervised Machine Learning algorithms, parameters of a model are learned by minimising a loss function of the true label and the prediction [63]. To train a NN the weights and biases of each layer are learned. In case of a multi-class classification, the training error is usually defined as cross-entropy:

$$Xent(t,y) = -\sum_{i \in classes} t_i \log y_i \tag{3.3}$$

where y is the output of the softmax function and t is the label where $t_i = 0$ unless i is the correct class, then $t_i = 1$.

To learn the weights and biases we use the back-prop algorithm. In its simplest form it uses gradient descent, an algorithm that approaches a local minimum step-wise. It can be shown that the gradient points in the direction of the steepest ascent [60]. If the objective function is the loss function, then a sufficiently small step in the opposite direction of the gradient of the loss with respect to the weights will therefore decrease the loss. The back-prop algorithm can be divided into a forward pass and a backward pass. During the forward pass the input is passed through all

layers

$$y = f_n(...f_2(f_1(x))). \tag{3.4}$$

During the backward pass the gradients are passed backwards from the output to the input and the weights are updated accordingly

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial f_n} \cdot ... \cdot \frac{\partial f_2}{\partial f_1} \cdot \frac{\partial f_1}{\partial x}. \tag{3.5}$$

More sophisticated optimisation methods like RMSProp [73] or ADAM [74] incorporate a dynamic learning rate for a more stable and fast convergence.

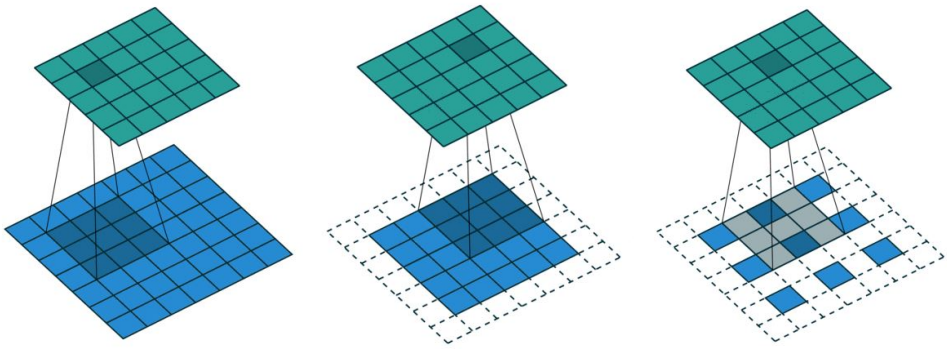
Training of a network can further be improved using dropout layers [75], that simulate missing features and prevent overfitting, or batch normalisation [76] that normalises the input to a layer for a more stable learning.

3.2 Convolutional Neural Networks

Convolutional Neural Networks [77] are designed to process two dimensional data. They preserve local spatial information but are invariant to the global location. In the convolutional layers of a neural network we *slide* filters across the input. Usually this input are images, but recently convolutional layer were also applied for instance to word embeddings in natural language processing [78]. The idea is the same as in image processing: context can be found independently of the location in a sentence, just like a car can be detected anywhere in the image.

Convolutions work as conventional hidden layers by computing the dot product of inputs with their respective weights and applying an activation function. The area that is convolved by a filter is called the *receptive field*, and defines how much spacial context is summarised in a pixel. In figure 3.2 (a) a convolutional layer with a filter size of 3 x 3 is depicted. Each pixel in the green image is the result of a convolution of the 3 x 3 filter with a 3 x 3 window in the original image centred around the pixel.

Variations of the convolutional layer include padding, as illustrated in figure



(a) (b) (c)

(a) Convolution. (b) Padding. (c) Transposed convolution.

Figure 3.2: Convolutional layer arithmetic [79].

3.2 (b), where the image is padded such that the output of the convolutional layer has the same size as the input. Further, strides allow to skip pixels and include more context without increasing the filter size and thus computational complexity. Deconvolutions as shown in 3.2 (c) pad the pixels such that the output is larger than the input. This increases the resolution of the image. Pooling layers decrease the resolution by averaging or taking the max around a window without applying a filter. The max pooling operation is illustrated in figure 3.3. For each of the four fields only the maximum element is preserved.

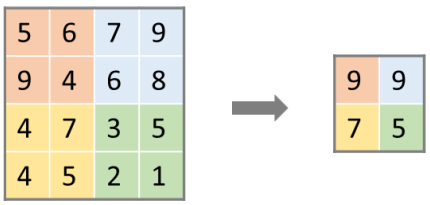


Figure 3.3: Max Pooling

Krizhevsyk *et al.* trained a CNN for image classification on the ImageNet dataset containing images of 22,000 categories [66]. In figure 3.4 the learned filters in the first layer of their network are depicted. Most filters in the first layer extract low level representations, information about frequency and orientation.

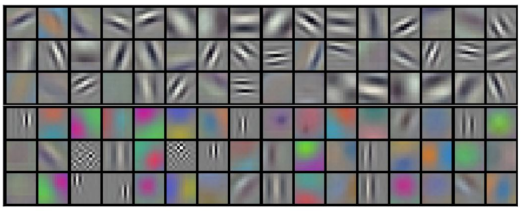


Figure 3.4: Krizhevsky's filters for classification [66]

3.3 A Review of Recent CNNs for Stereo Matching

Recently a number of approaches to the stereo matching problem using CNNs were suggested. Many of them treat the correspondence problem as classification problem, others as regression problem with further refinement after the CNN. The most distinct proposals of the last two years are discussed this section.

Zagoruyko *et al*. described and compared different Convolutional Neural Net (CNN) architectures used to learn similarity functions from two raw image patches [80] in 2015. Their intention was to improve matching for numerous applications like for example classification or image recognition. The training set consisted of pairs of matching and non-matching patches with binary labels. The considered architectures included Siamese structures, where both patches were processed independently but with shared weights before being concatenated and jointly mapped by a fully connected layer. Effectively this approach first computed a descriptor for each image and subsequently adopted similarity to these descriptors. Alternatively, they proposed to process both patches simultaneously by treating the image pair as 2-channel input to the CNN. It was shown that learning similarity measures directly from the images with either one of the proposed architectures could significantly outperform manually-designed descriptors like SIFT.

Around the same time, Žbontar and LeCun exploited the Siamese structure to compare images patches for the purpose of stereo matching [81, 82]. They proposed an architecture that was significantly faster with only minor losses in accuracy. In particular, they replaced the concatenation step, which was usually followed by several fully connected layers by simply applying the dot product as a measure of similarity. Again, binary labels were employed, with the difference that for each training

patch there was both, a matching and a not matching example. A training loss of zero was assigned, if the similarity of matching example exceeded the similarity of the negative example by a fixed threshold. Employing the Siamese architecture although the 2-channel approach performed significantly better in Zagoruyko's experiments [80], further decreased computation time. Although the similarity computation still needed to be executed for each possible disparity, in the Siamese NN, the computation of the descriptors only needed to be done once for each centre pixel. Žbontar *et al.* further proposed the application of several optional post-processing steps. They used a combination of

- Cross-based cost aggregation, where the matching costs are averaged over a fixed window.
- Semi-global matching as proposed by Hirschmuller (see 2.4.1 [46]).
- A left-right consistency check, using the right image as reference and comparing results.
- Subpixel enhancement, that increases the resolution of the pixelwise prediction.
- A median and bilateral filter that as a last step smooth the disparity map.

Luo *et al.* [83] also proposed to use the inner product as similarity measure for efficiency reasons. They expanded Žbontar's work by outputting a probability distribution over all possible disparities, which were treated as different classes. The standard procedure until then was to compare two patches of the same size for each disparity independently and picking the patch that yields the highest similarity score. In Luo's new approach, the right image incorporated the entire search space which lies on a horizontal line for rectified images due to the epipolar constraint. This allowed a joint estimation of the similarity for different disparities and thus the exploitation of a softmax function used to train the network with cross-entropy loss.

Park and Lee [84], as well as Dosovitskiy *et al.* [85] attempted to increase the receptive field of their networks. Park *et al.* used pooling layers with differ-

ent strides and kernel sizes and concatenated their output. By simply employing both, pooling layers with small and large window sizes, they avoided the trade-off between large windows that lose details and small windows that lose spatial information. Dosovitskiy *et al.* estimated optical flow with CNNs. They used transposed convolutions and bilinear upsampling after their pooling layers in order to avoid the blurring of small details and to achieve a dense per pixel prediction.

Zoox *et al.* also trained a CNN that processes the input images independently but with shared weights in the first layers to tackle the stereo matching problem [86]. However, instead of computing the disparity map as a intermediate step, the output of the network was a predicted new view. The advantage was that no labelling for the training set was required. The network was trained on a set of posed images by leaving one of the images out and using it as the label. In that way, the NN learned to reproduce a view on the scenery. Although this method was more robust to common problems of image rendering, like tearing and elimination of fine structure in self-occluding objects, it did not provide geometrical information needed for the reconstruction.

Kendall *et al.* took the end-to-end approach to a new level in March 2017 and achieved the best results on the KITTI data set so far [87]. Their approach was to solve the stereo vision problem with DL incorporating understanding of stereo geometry, as opposed to all previous approaches. That far the DL approaches had focused on representing unary costs, in other words step (1) of Scharstein's taxonomy. Kendall *et al.* incorporated steps (1) - (4) in a single NN, with the result that no post-processing or regularization steps were needed. The key to this approach was that a fully differentiable cost volume was used that retained the feature dimension and passed it through the network. In contrast, in former approaches the feature representation was eliminated by a dot product and replaced by a disparity representation, which was the output of the NN. Kendall argued that retaining the feature dimension allowed to explicitly reason about geometry while learning the entire model end-to-end. The input to the network are stereo pairs of images. Similar to previous approaches the first layers in their NN were Siamese convolutions

with shared weights. Next a disparity cost volume is obtained by concatenating the unary features of both images. 3D convolutions are then used to learn context by learning feature representations from the height, width and disparity dimensions. The resulting increase in computational complexity is regulated by sub-sampling. Instead of using an argmin operation to output the disparity with the lowest cost, a differentiable and smoothing *soft argmin* operation is defined. The authors showed that their network reasons about local geometry using wider contextual information than the standard 9x9 windows, without being set to a certain window size.

Chapter 4

Model Design

In this chapter we analyse the potential of different CNN architectures to predict the disparity map of a scenes captured with endoscopes during surgery. We first describe the training dataset and challenges that arise from the differences between the scenes in the training set, and the surgical scenes we apply the models to. We then conclude why some architectures could outperform others and introduce our contribution which is an improved CNN architecture. We describe in detail the different models we evaluate and compare in this thesis. The result of this comparison will we discussed in chapter 5.

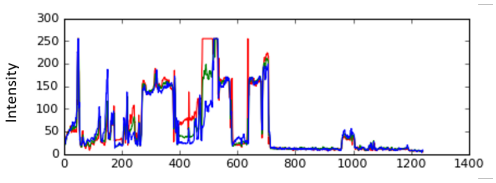
4.1 The Training Dataset

As to our knowledge there is no dataset with a ground truth disparity for images from a surgical context, the models described in this chapter are trained on a data set depicting street and city scenes. We use the KITTI stereo 2015 dataset [88], which consists of stereo pairs and a ground truth disparity of 200 scenes that was acquired by means of a laser scanner and a GPS localization system. It was introduced by Menze and Geiger and is used as a common benchmark for depth estimation.

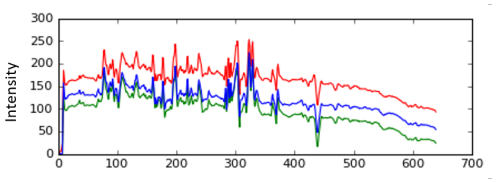
An example from the data set can be seen in figure 4.1. Each training example contains a left image, a right image and a disparity ground truth. The rectified and cropped RGB images are approximately 376 x 1244 pixels large. It can be observed in figure 4.1 that large parts of the images do not have a ground truth, for example the sky or occluded areas.

Figure 4.1: Left view, right view and the ground truth depth.

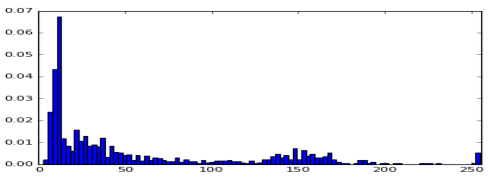
The dataset used for training the models is different from the intended application. In the images of city scenes there are many edges such as signs, road marking, buildings etc. The colon on the other hand, has no sharp edges and less texture. It can be observed in figure 4.2 that the RGB intensities of images of tissue are



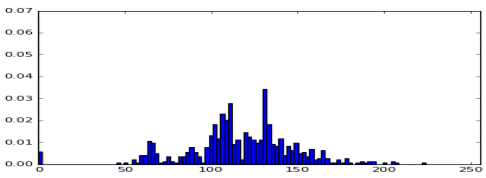
RGB intensities along an epipolar line of a typical image from the KITTI data set



RGB intensities along an epipolar line of an image of the inner surface of a colon.

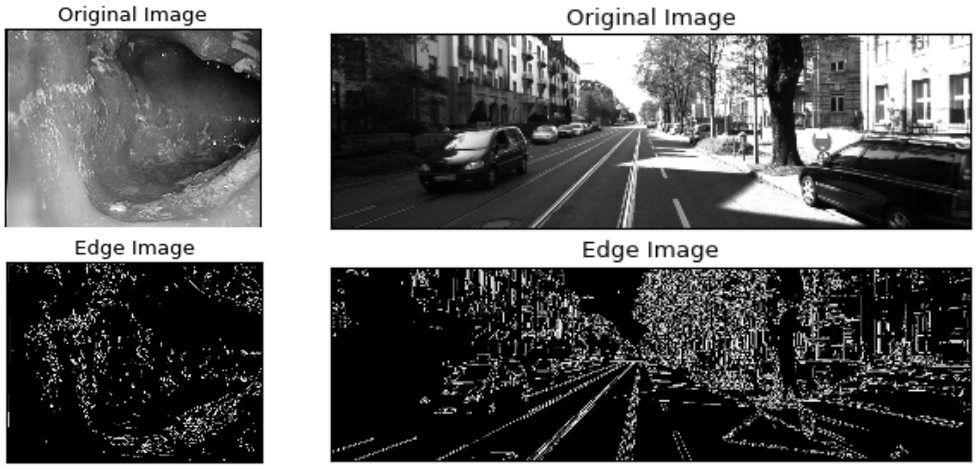


Histogram of the intensities of the red channel along the same epipolar line as above. Values are spread among almost all intensities of the spectrum (0,255).



Histogram of the intensities of the red channel along the same epipolar line as above. Values are concentrated around the interval (100,150).

Figure 4.2: Comparison of the RGB intensities of two images along an epipolar line.



(a) (b)

(a) Canny edge detector applied to an image of a colon phantom. Few edges are detected in the shady areas in the back of the tube. The detected edges are rather scattered. (b) Canny edge detector applied to an image of the KITTI data set. Only few edges are detected in shady areas as well. However, the edges that are detected show straight lines around street markings, windows and buildings. Even the lights, wheels and the plate of the car can be detected.

Figure 4.3: Canny edge detector applied to different images.

typically unimodal and less discriminative than those of images from the KITTI dataset. As a result, pixels along the search space are more similar leading to a potentially flatter and multimodal probability distribution over the disparities, making the model less certain.

The different textures of the two scene types are also illustrated in figure 4.3, where a Canny edge detector is applied to both data sets. The comparison gives an idea how filters could more easily extract information from images with sharp edges than from images of human tissue.

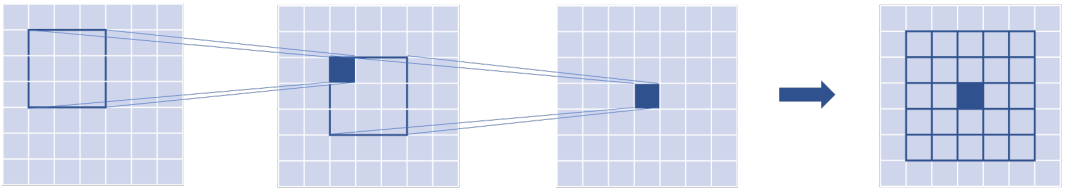
Early approaches to the stereo matching problem have aimed to find these edges in images and match them as discussed in chapter 2.4.1. Although Convolutional Neural Nets use a more complex feature representation they also tend to extract information about edges in the first convolutional layer as we have seen in figure 3.4. A network trained on images with distinct edges could therefore rely on the presence of those and as a result be less accurate on images that have less texture.

Therefore, we acknowledge that there is no guarantee that models verified on the KITTI dataset will be suited for applications in a surgical context. However, we expect that the insights we obtain from the KITTI dataset will have informative value for the reconstruction of a colon. In fact, our results show that the better a model performs on the KITTI data, the better its performance on medical data.

4.2 Increasing the Receptive Field of CNNs

Our goal is to derive the model architecture that promises best results on the correspondence problem in applications to surgery by investigating how different models handle different scenes. To this end we observe the predictions made by different models on areas in the images whose depth is typically hard to predict in medical imaging. We have discussed that a common difficulty in this context is the lack of edges and texture. Another challenge are reflections that arise from the very bright light source of the endoscope that is necessary to illuminate the abdomen. In an image reflections will be captured as an area of purely white pixels that are not matchable independently of one another. How can we decide which of the white pixels in the left image should be matched to which of the white pixels in the right image? Further, reflections depend on the angle. The two stereo cameras will therefore observe different scenes. But by looking at the context, the pixels around the reflections, the model can draw conclusions. We call the area the model ”looks at” when classifying a pixel the receptive field. We widen it successively, reporting the performance of the different models. The conclusions we draw can then be applied to stereo images of the colon. We do not expect that the results are directly applicable to surgical images, rather we experimentally verify the behaviour on colon data by qualitatively examining the results in chapter 5.

We believe that widening the receptive field will enable the model of a better contextual understanding of endoscopic scenes and improve the matching in textureless and reflective areas like the colon. We experiment with different depths of networks and different number of down-sampling layers that widen the receptive field to find the optimal width. We vary the depth of the network between 4, 7, and



The centre pixel in the third image is the result of the 3x3 convolution in the second image. Every pixel of this 3x3 receptive field is the result of another 3x3 convolution (first image). Effectively, the receptive field of a pixel that is convoluted twice with 3x3 filters is therefore that of a 5x5 convolution as depicted in the fourth image.

Figure 4.4: Receptive field of two 3x3 convolutions.

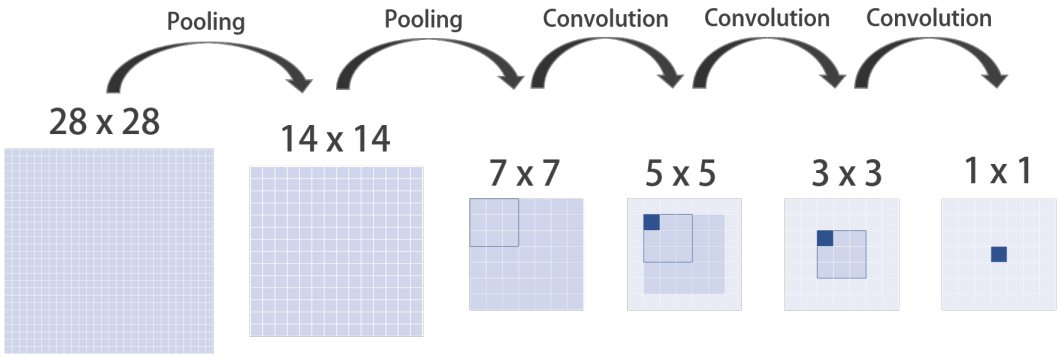
9 layers and use 0, 2, or 3 down-sampling layers. We further observe the validation error to find the optimal number of epochs. We emphasise, that the goal of this research is to optimise the initial matching only. Further refinement and post-processing of the predicted disparities is outside the scope of this work and remains to be investigated.

We increase the receptive field with two operations: Convolutional layers and pooling layers.

Convolutional Layers Figure 4.4 illustrates how stacking several convolutional layers increases the receptive field. Applying a 3 x 3 filter to an image condenses context of a 3 x 3 area in one pixel. Applying a second convolution incorporates information from another 3 x 3 field that overlaps with the first one. In this manner the receptive field in the k^{th} layer can be calculated as $(3+2\cdot k)$ x $(3+2\cdot k)$.

Figure 4.4 illustrates why two subsequently applied convolutional filters of size 3 x 3 impose the same theoretical receptive field as one 5 x 5 filter. Convoluting a convoluted pixel successively widens the receptive field. Stacking a third layer of 3 x 3 convolutions yields a receptive field of 7 x 7. Although the receptive fields are equivalent in both described cases, the different architectures are not to be seen as equal. Simonyan and Zisserman demonstrate in [89] that 3 convolutional layers with a filter size of 3 x 3 are to be preferred over one convolutional layer with filters of size 7 x 7. They argue that, firstly, stacking three convolutional layers incorporates three non-linear activations

instead of one, allowing the model to learn a more discriminative decision function. Secondly, they exercise how one convolutional layer with 7 x 7 filters would need 81% more parameters than three 3 x 3 convolutional filters. The authors view this reduction in parameters as imposing a regularisation on the 7 x 7 convolutional filters, forcing them to learn the same representation with less weights. We therefore chose to let the model learn the best representation and incorporate kernels of size 3 x 3 pixels only.



The model down-samples patches of size 28 x 28 to a size of 7 x 7 in order to increase its receptive field. Similar to the illustration in figure 4.4 the 3 convolutions increase the receptive field from 3 x 3 to 7 x 7 pixels. The two pooling layers have a stride of 2 and thus double the receptive field twice yielding a receptive field of 28 x 28 pixels. Additional convolutional layers between the pooling layers are not illustrated here.

Figure 4.5: Illustration of the receptive field of a model with 2 down-sampling layers.

Pooling Layers The second operation, max pooling layers, was introduced in chapter 3.2. The pooling layers used in our architecture condensate information of an image of size 28 x 28 with a stride of 2 and 2 x 2 kernels in 14 x 14 pixels. The downside of pooling layers is that they come hand in hand with a loss of detail. Therefore, we apply the same number of transposed convolution layers that up-sample the feature representation of the input images. While Park and Lee argue that pooling layers can lose small detail for good [84], we argue that the textureless regions like tissue will benefit from a wider receptive field. Besides the increase in the receptive field this approach also serves as a regularization operation, finding a lower dimensional representation of previous layers preventing from overfitting and allowing faster

processing. Park and Lee apply different pooling layers (different strides and window sizes) to the same feature representation concatenating all the outputs. We see this as implicitly applying exhaustive search among all possible pooling operations, that could be avoided.

We apply several of these operations to our model architecture. A variant with two down-sampling layers is depicted in figure 4.5. Further convolutional layers between the pooling layers are not depicted in this example because the increasing receptive field caused by those, is overshadowed by the pooling layers. We experiment with different receptive fields to find the best trade-off between increase in receptive field and loss in detail and report the results in chapter 5.

4.3 Network Architectures

We use a Siamese architecture in our model as proposed by Luo *et al.* [83] and take advantage of their approach to use an inner product as similarity measure. This approach reduces computational complexity in two ways:

- The Siamese structure allows to compute a feature representation of an image once for all disparities instead of recomputing it for every single disparity value (129 in our case) for each pixel.

- The inner product speeds up computation significantly compared to several fully connected layers [82] or correlation layers followed by several convolutional layers [85], which are proposed instead. A computationally cheap similarity measure allows a simpler study of the effect of an increased receptive field on the accuracy of a model. Further, this approach allows input images of different sizes as opposed to fully connected layers.

We tackle the matching problem as classification problem and use CNNs that classify each pixel of an image as a disparity between zero and the maximal disparity ($maxDisp$) 128. Because we distinguish multiple classes we use cross-entropy loss. We only use integer disparities and do not calculate subpixel disparities.

As labels are sparse we only backpropagate pixels whose groundtruth is non-zero. For a more stable and faster convergence we apply batch normalisation before each activation (ReLU) as proposed in [76]. The weights of both branches of the Siamese network are shared such that both images, left and right, are transformed equally. This is a requirement for the inner product layer as similarity measure. In our network we randomly initialize weights once and have both branches of the network access the relevant weights. During backpropagation the gradients from both branches are used to update the weights.

We proceed similar to Luo's approach and stepwise increase the receptive field reporting the results. However, opposed to their work, we incorporate several pooling layers and transposed convolutional layers to investigate the effect of down-sampling. Similar to Dosovitskiy's approach to Optical Flow estimation in [85] we use transposed convolutional layers after the aggregation through convolutional and pooling layers to refine the feature representation. However, we up-sample the features to the original image resolution to get a dense per pixel prediction of the depth. While Dosovitskiy's approach included bilinear upsampling we decided to let the network learn the best refinement and only incorporated transposed convolutional layers.

Table 4.1 gives an overview over the six different models we train and evaluate. We gradually include more convolutional layers and more pooling layers in order to investigate their impact on the results. The models without pooling layers (M2, M4) are adoptions of Luo's [83] model with different numbers of layers. Model M4 is an implementation of their 19 x 19 receptive field model.

Instead of training on whole images we randomly sample small patches of 28 x 28 or 56 x 56 pixels for models with a receptive field of $\geq$ 28 x 28, from the left image as in [83]. From the right image we extract the corresponding 28 x 156 pixel patch, or 56 x 184 patch respectively.

Similar to Luo's approach [83] the right image patch is wider than the left one. As the images are rectified, the search space lies along a horizontal line only. Therefore the height of both patches is the same, but the width of the right patch equals

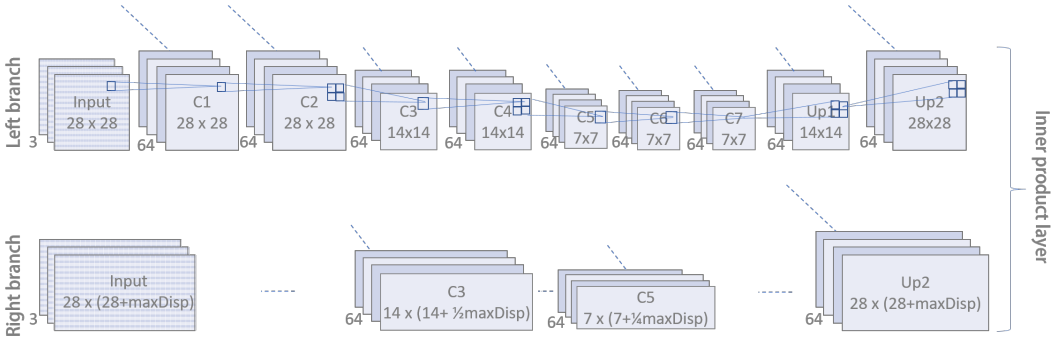
Overview over different model architectures

c	M1	c	M2 [83]	c	M3
1	conv3 + BN + ReLU	1	conv3 + BN + ReLU	1	conv3 + BN + ReLU
2	conv3 + BN + ReLU	2	conv3 + BN + ReLU	2	conv3 + BN + ReLU
	pool 1	3	conv3 + BN + ReLU		pool 1
3	conv3 + BN + ReLU	4	conv3 + BN + ReLU	3	conv3 + BN + ReLU
4	conv3	5	conv3 + BN + ReLU	4	conv3 + BN + ReLU
	deconv 1	6	conv3 + BN + ReLU		pool 2
		7	conv3	5	conv3 + BN + ReLU
			softmax	6	conv3 + BN + ReLU
				7	conv3
					deconv 1
					deconv 2
					softmax

c	M4 [83]	c	M5	c	M6
1	conv3 + BN + ReLU	1	conv3 + BN + ReLU	1	conv3 + BN + ReLU
2	conv3 + BN + ReLU	2	conv3 + BN + ReLU	2	conv3 + BN + ReLU
3	conv3 + BN + ReLU	3	conv3 + BN + ReLU		pool 1
4	conv3 + BN + ReLU		pool 1	3	conv3 + BN + ReLU
5	conv3 + BN + ReLU	4	conv3 + BN + ReLU	4	conv3 + BN + ReLU
6	conv3 + BN + ReLU	5	conv3 + BN + ReLU		pool 2
7	conv3 + BN + ReLU	6	conv3 + BN + ReLU	5	conv3 + BN + ReLU
8	conv3 + BN + ReLU		pool 2	6	conv3 + BN + ReLU
9	conv3	7	conv3 + BN + ReLU		pool 3
		8	conv3 + BN + ReLU	7	conv3 + BN + ReLU
		9	conv3 + BN + ReLU	8	conv3 + BN + ReLU
			deconv 1	9	conv3 + BN + ReLU
			deconv 2		deconv 1
			softmax		deconv 2
					deconv 3
					softmax

Abbreviations: c = no. of convolutional layer, M = model, conv3 = convolutional layer with 3x3 filter, BN = batch normalisation, ReLU = rectified linear unit, pool # = pooling layer no. #, deconv #= transposed convolutional layer (up-sampling layer) no. #.
We use 64 features in each convolutional and transposed convolutional layer and a stride of 1. For the pooling layers we use a stride of 2. We remove the ReLU from the last layer to keep information from negative values.

Table 4.1: Comparison of the architecture of six different models.



Abbreviations: C = convolutional layer, Up = upsampling layer.
Architecture of model M3 with 7 convolutional layers, two pooling layers and two upsampling layers. The right branch is only depicted partially.

Figure 4.6: Convolutions

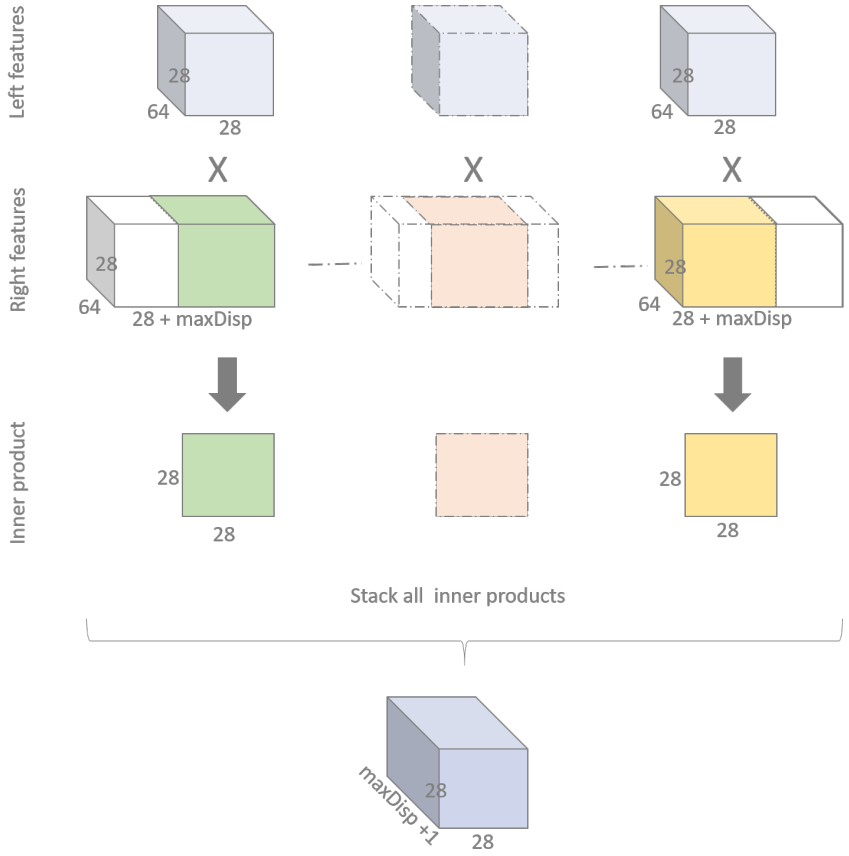
the width of the image plus the maximum disparity between a pixel in the left image and a candidate pixel in the right image. By limiting the search space to one dimension we reduce computational complexity significantly and disregard implausible matches from begin with. However, the downside is that the rectification of images is not trivial and is highly sensitive to the calibration of the cameras, which imposes an additional source of noise [16]. The architecture of one representative model during training is depicted in figure 4.6. The inner product layer is illustrated in figure 4.7.

The inner product of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is defined as

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y} = |\mathbf{x}||\mathbf{y}| \cos(\theta) \tag{4.1}$$

where θ is the angle between the two vectors. Note that $|cos(\theta)| \in [0,1]$, and $\cos(0) = 1$ and $\cos(90) = 0$.

If the lengths of the vectors are fixed then their inner product is therefore largest when $\theta = 0$ or in other words when one vector is a multiple of the other. The inner product is smallest, when $\theta = 90$, that is, when the vectors are orthogonal to one another. Intuitively, the inner product is largest when the vectors are most similar. Therefore, the inner product is a convenient measure of similarity and we focus on finding the best suited feature representation (as a vector) of a pixel.



The product layer computes the inner product between the feature representation of each pixel of the left image with the feature representation of the corresponding pixel in the right image shifted by each possible disparity. As the disparity can be zero there are $maxDisp+1$ different multiplications. Stacking these similarity matrices yields a volume with a value for each pixel in the reference image and each possible disparity.

Figure 4.7: Product layer

Our inner product layer implicitly slides the left patch over the longer right patch and calculates the inner product as a measure of similarity at each position. The position that yields the highest similarity corresponds to the most likely disparity of a pixel. The procedure is illustrated in figure 4.7.

Let $\mathscr{D}$ be the set of all possible disparities $d = 0,\ldots,maxDisp$ with $|\mathscr{D}| = maxDisp+1$. Let further $f(I^L_{i,j}) \in \mathbb{R}^{64}$ denote the feature representation of pixel $p_{i,j}$ in the left image and let $f(I^R_{i,j}) \in \mathbb{R}^{64}$ denote the feature representation of $p_{i,j}$ in the right image that is imposed by our CNN. Similar to [83, 82] we take the Winner-take-all approach and discard global approaches as proposed

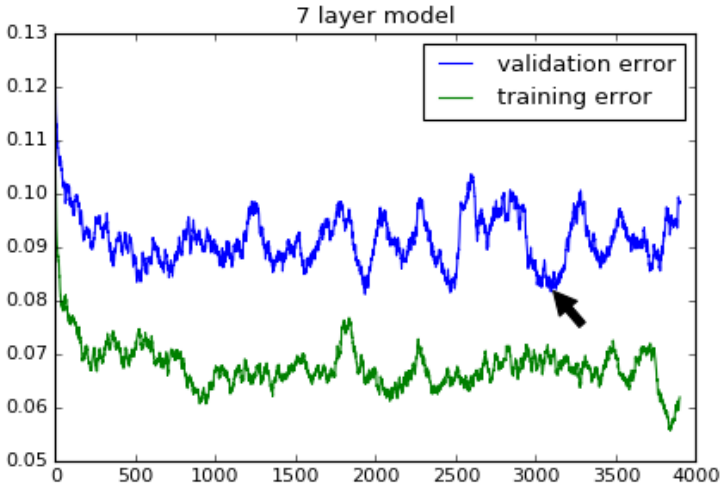


Figure 4.8: Validation error on random batches.

in [90, 87]. The reasons are that firstly, we focus on the initial matching only in this dissertation, and secondly, the larger receptive field already incorporates information from a large window, even though not on a global basis. As a result, the predicted disparity $d_{i,j}$ of each pixel $p_{i,j}$ for all $i, j = 1, ..., imSize$ is obtained as

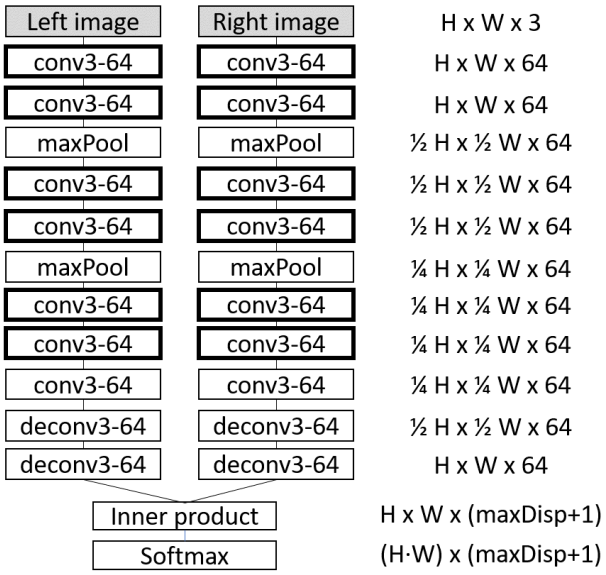
$$d_{i,j} = \max_{d} \sum_{k=1}^{64} f(I^{L}_{i,j,k}) \cdot f(I^{R}_{i,j-d,k}) \tag{4.2}$$

where "$\cdot$" describes the multiplication of two scalars.

4.4 Training and Testing

We train different models and evaluate each one on the same, randomly selected validation set. Figure 4.8 indicates that after 300k iterations there is no further improvement in the validation error, while the training error keeps decreasing. We therefore decide to train all models with a batch size of 32 for 300k iterations. An exception is M6 that is trained on patches of size 56 x 56. Due to memory constraints of the used GPUs M6 is trained on batches of 8.

While we trained on random patches of a fixed size during training, we predict the entire image during testing. The architecture of our network remains the same, however the input to our networks is now larger. The architecture of an example



Abbreviations: H = height, W = width, maxPool = max pooling layer, conv/deconv<receptive field size>-<number of channels> = (de)convolutional layer with specifications (adapted from [89]). Layers with bold frame apply batch normalisation and ReLUs.
During Prediction the input to the network is an entire image. The dimensions of the feature representation in each layer are depicted next to the network architecture.

Figure 4.9: Network Architecture for Prediction

model (M3) and the dimensions of each layer are illustrated in figure 4.9.

4.5 The Colonoscopy Dataset

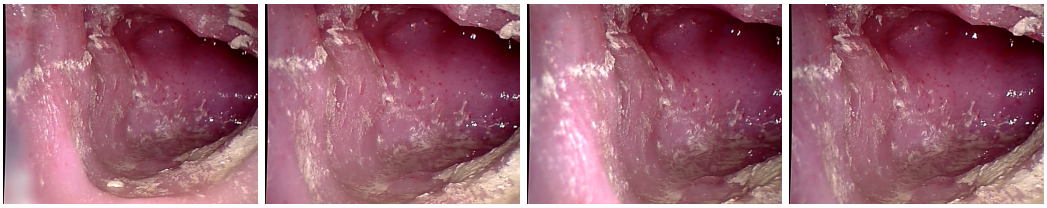


Figure 4.10: Four frames of the test video of the colon phantom.

The dataset was acquired from a phantom made of PVA-C and talcum powder. First, a mould was 3D printed using a CT scan of a human colon. The lumen was filled with clay to gain the characteristic folds of the colon as a negative mold. This negative mold was supported inside a container that was then filled with the cryo-gel and under went the freeze thaw cycle [91] to create the phantom. The

clay was removed after to leave a completed phantom. As the stereo endoscope of the DaVinci surgical system is not bendable we can only observe an area at the beginning of the phantom of roughly 8 cm depth. We obtained a video and extracted four frames for which we predict the 3D coordinates.

4.6 Post-processing and Triangulation

From the disparity map we can read off the match of each pixel $p^L_{i,j}$ in the left image as pixel $p^R_{i,j-d_{i,j}}$ in the right image. We explicitly discard post-processing of the initial matches; however, we discard those matches whose output of the softmax layer does not exceed a certainty threshold $t \in [0,1)$. In other words, we ignore matches for which the probability of a match is too low. This reduces noise and outliers considerably and the number of remaining matched pixels gives a good idea of the density of the matched points. Triangulation is performed using *traingulate* from MATLAB's Computer Vision System Toolbox which is an implementation of the procedure desribed in chapter 2.3. We finally use an iterative closest point (ICP) algorithm to align four overlapping point clouds from several frames. We apply MATLAB's *pcregrigid* function according to [92] to register triangulated points from the four frames depicted in figure 4.10. This function estimates a 3D surface for both clouds and minimises a distance measure between them.

Chapter 5

Experiments and Results

In the previous chapter we introduced different models with different theoretical properties, strengths an weaknesses. In this chapter we validate the models on different datasets. We first validate our models on the KITTI dataset to compare our results on a common benchmark. We then apply the promising models to stereo images of the colon phantom and conduct a thorough qualitative evaluation of the resulting disparity maps. We triangulate our matched points and compare the 3D model to a computed tomography (CT) scan of the phantom to present a quantitative evaluation which underpins the results of the qualitative evaluation. Lastly, we apply our best model to real data of porcine intestines. Although we do not have a ground truth we will see that we obtain a dense and very detailed reconstruction of an intraoperative scene.

5.1 Evaluation Metrics

In this section we briefly introduce the evaluation metrics used for comparison in the remainder of this chapter.

- 2, 3 and 5 pixel error as measure of accuracy. Depending on the required degree of accuracy models can rank differently well. The n pixel error (n PE) is defined as ratio of correctly, e.g. up to a deviation of n pixels from the true disparity, classified pixels and the total number N of pixels that have a ground

truth:

$$n\text{ PE} = \frac{1}{N}\sum_i \sum_j \mathbb{1}_{\{|d_{i,j}^{true} - d_{i,j}^{pred}| \leq n\}} \tag{5.1}$$

- Certainty as maximum of the softmax output of a pixel or in other words the maximum probability over all densities for pixel $p_{i,j}$:

$$\max_{d_{i,j}} P(d_{i,j}). \tag{5.2}$$

A model with steep and unimodal probability distributions is less unambiguous.

- Matching density as ratio between number of matched pixels and total number of pixels:

$$\text{matching density} = \frac{\text{number of matched pixels}}{\text{number of pixels}}. \tag{5.3}$$

We use this metric when we do not have a ground truth for the correct disparity. We call pixels *matched* if their certainty exceeds a threshold t:

$$dens(t) = \frac{1}{N}\sum_i \sum_j \mathbb{1}_{\{P(d_{i,j}^{pred}) > t\}} \tag{5.4}$$

- Qualitative metrics: Description of shapes and areas. Ranking of noise and density levels of different models.

- The reconstruction error as mean absolute distance between a compared cloud $\mathscr{C}_{comp}$ and a reference cloud $\mathscr{C}_{ref}$:

$$\varepsilon = \frac{1}{|\mathscr{C}_{comp}|} \sum_{p_i \in \mathscr{C}_{comp}} ||p_i - p_i^N||_2 \tag{5.5}$$

where

$$p_i^N = \min_{p_j \in \mathscr{C}_{ref}} ||p_i - p_j||_2 \tag{5.6}$$

is the nearest neighbour of a 3D point p_i in the reference cloud and the 2-Norm denotes the euclidean distance.

- Root mean square (RMS), similar to mean distance but more robust:

$$RMS = \sqrt{\frac{1}{|\mathscr{C}_{comp}|} \sum_{p_i \in \mathscr{C}_{comp}} ||p_i - p_i^N||_2^2} \tag{5.7}$$

5.2 Results on the KITTI Dataset

To compare different CNN architectures we first evaluate our models on the KITTI dataset. The performance was measured as n pixel error. The results are stated in table 5.1.

Model	convs	poolings	RF	2 PE	3 PE	5 PE
M1	4	1	10 x 10	11.88	9.51	7.70
M2 [83]	7	0	15 x 15	11.72	9.45	7.66
M3	7	2	28 x 28	**9.82**	**7.37**	**5.83**
M4 [83]	9	0	19 x 19	11.18	8.95	7.18
M5	9	2	28 x 28	10.03	7.48	**5.83**
M6	9	3	56 x 56	12.34	8.47	6.18

Abbreviations: RF = receptive field, PE = pixel error, convs = number of convolutional layers, poolings = number of pooling layers.

Table 5.1: 2, 3 and 5 pixel error of different models on the KITTI 2015 dataset.

The state of the art 3 pixel error is 2.87% [87]. However, this result is not directly comparable to our approach. While Kendall *et al.* trained an end-to-end model we focussed on improving solely the feature extraction, in other words part (1) of Scharstein's taxonomy. Our results must therefore be compared after the matching task without further post-processing. A benchmark for this are the results of Luo *et al.* [83]. They reported a 3 pixel error of 8.95% for their 9 layer model with a receptive field of 19 x 19 and our implementation of their model (M4) yields

the same error.

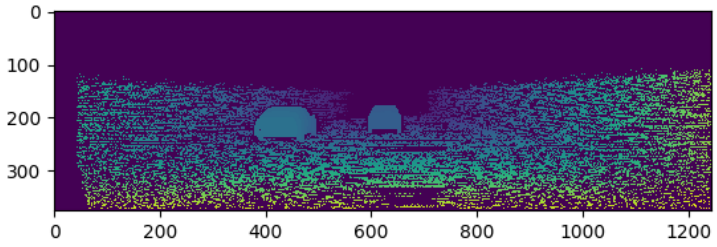
Comparing the models in table 5.1 we note that model M3, our 7 layer model with two pooling layers, yields the best results on the KITTI 2015 dataset. Model M1, the most shallow model with the smallest receptive field, performs worst.

M4 has the same architecture as M2 with two additional convolutional layers. As M4 performs better than M2 a deeper network seems beneficial; however, this argument is not valid in general. M5 has the same architecture as M3 with two additional convolutional layers, but we observe that M5 yields a higher error than M3, from which we conclude that depth is not the main determining factor in the performance of the matching. Comparing M3 to M2 and M5 to M4 it becomes apparent that it is rather the receptive field that determines performance. M3 and M2, and M5 and M4, respectively, have the same depth but very different receptive fields. We conclude that the increase in the receptive field is crucial for the accuracy of the matching algorithm. However, the relation is not linear. M6, that has a considerably larger receptive field than M3, yields a lower accuracy on the validation set. One possible explanation is that due to the down-sampling details are lost or blurred.

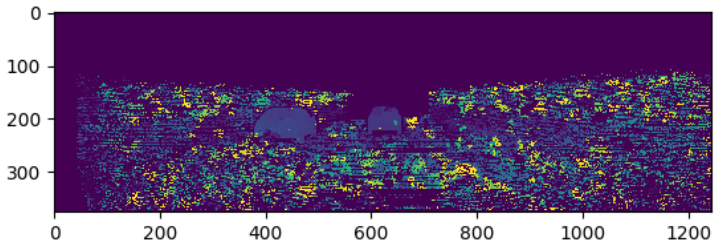
In the following we look at two examples and give some explanations for the different behaviour of the models. We compare the 7 layer model without pooling (M2) and the 7 layer model with two pooling layers (M3). M3 outperforms M2 on every validation image. We evaluate both models on two examples of the validation set: One where both have a very high error, 77% and 84% respectively, and one where both have a very low error, 2% and 4%, respectively.

The photo in figure 5.1 was taken in a tunnel where the lightening conditions prevent from a successful estimation of depth. All models report an error of at least 70%. Comparing the prediction of M3 to the prediction of M2 gives an intuition of how the models deal with scenes that are hard to predict. While the depths predicted by M2 look rather randomly scattered, those predicted by M3 are more concentrated yielding a patched pattern. This is coherent with the intuition we gave for introducing the different models. The model with pooling layers has a larger receptive field and tends to compare a pixel to the other pixels in this field, while the

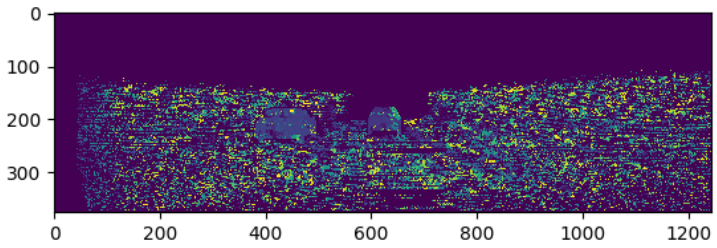
(a)



(b)



(c)



(d)

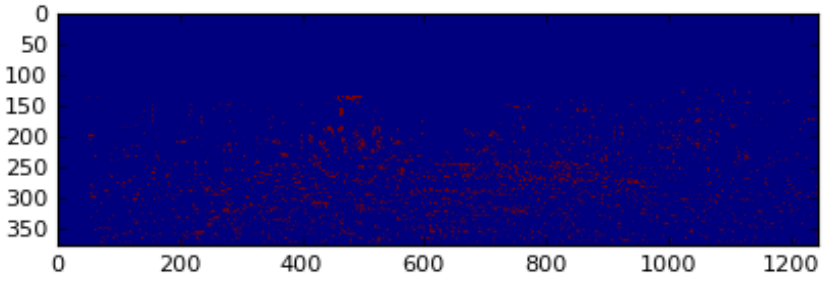
(a) Left view on test image. (b) Ground truth for depth. (c) Prediction of M3. (d) Prediction of M2.

Figure 5.1: Comparison of predictions of M2 and M3 of a test image.

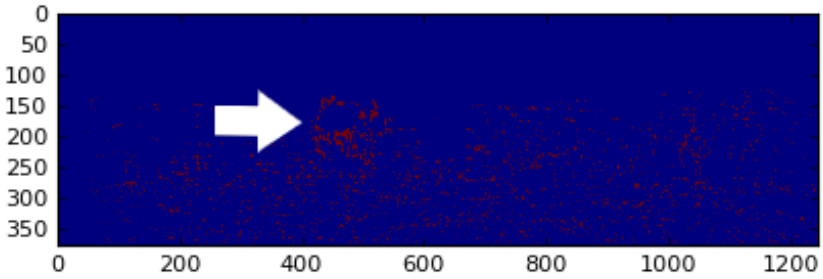
model without pooling layers has a small receptive field and therefore the prediction of one pixel depends less on its surroundings.

The second example shown in figure 5.2 illustrates the behaviour of the mod-

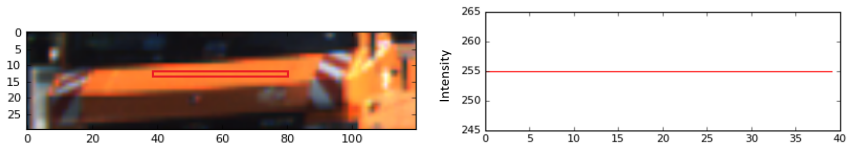
(a)



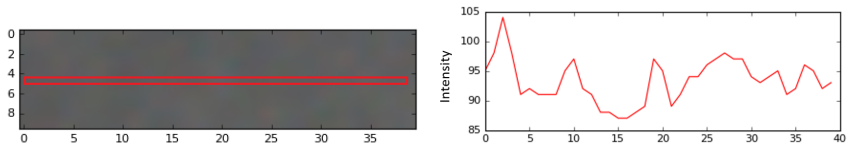
(b)



(c)



(d)



(e)

(a) Left view on the test scene. (b) Pixels M2 classifies correctly, but not M3. (c) Pixels M3 classifies correctly, but not M2. (d) Zoom into the orange part of the car and corresponding red channel. (e) Zoom into street and corresponding red channel.

Figure 5.2: Example: M3 outperforms M2.

els on surfaces without texture. The orange car has a bright neon colour that has no visible texture. Additionally, the shape of the orange parts lies horizontally on the epipolar lines, and thus along the search space. This yields a very flat probability distribution, effectively every orange pixel in the right image is equally likely to match any orange pixel in the left image. The second image in figure 5.2 depicts the pixels on which M3 fails, but which M2 predicts correctly. The third image depicts the opposite case. While there is no obvious pattern in the first case, one can see a systematic error in the predictions of M2. The errors of the model without pooling layers lie especially along the orange parts of the car. The model with the smaller receptive field is not capable of determining similarity, because it does not include enough context. Model 3 incorporates this context, enabling the model to distinguish orange pixel along the same epipolar line from one another by considering its neighbours.

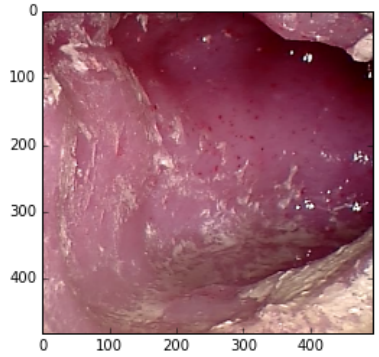
The results observed on the KITTI set are not transferable to endoscopic images. However, they give an idea about the behaviour of different architectures and imply from our perspective that a model with 2 or 3 subsampling layers will widen the receptive field and enable a better understanding of scenes with no edges.

In the following section we will investigate into the performance of models M2, M3 and M6 to compare the impact of zero, two or three pooling layers, respectively.

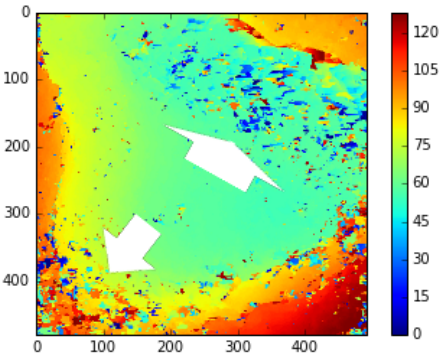
5.3 Reconstruction of the Colon

We apply models M2, M3 and M6 to the video we acquired of the colon. We do not change any parameters, but use the same reconstruction as implemented for the KITTI set. In this section we first conduct a qualitative evaluation of the generated disparity maps. Subsequently, we compare the resulting 3D models to the ground truth in the form of a CT scan.

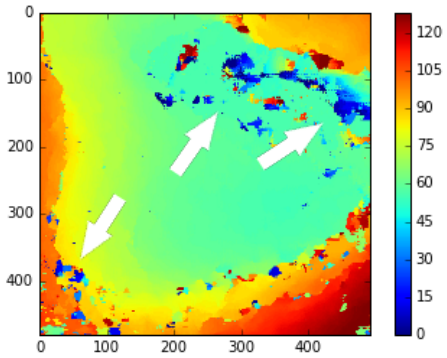
We begin our analysis with a look at the disparity map in figure 5.3 where some obvious outliers are indicated by white arrows. For these pixels a disparity of less than 30 is predicted, which would correspond to a depth of 15 cm with equation 2.1, a focal length of 843 pixels and a baseline of 5.35 mm. According to the



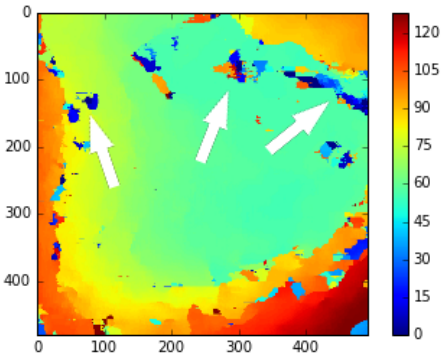
Original image of the phantom.



Disparity map of model M2.



Disparity map of model M3.



Disparity map of model M6.

Figure 5.3: Comparison of disparity maps of different models.

geometry of the CT model, this depth is not possible. These errors behave similar to the observations we made on the KITTI dataset. Model M2 (no pooling layers) yields scattered errors throughout the whole image. The errors of M6 (three pooling layers) are very patchy, there are six relatively big patches that are obviously wrong, but besides this the disparity image is very smooth. Model M3 (two pooling layers) yields errors that behave somewhat in between the two extremes. However, based on the disparity maps alone we cannot prefer one model over the others.

To get an idea why the predicted disparities of some pixels are so far off, we determine pixels that are classified incorrectly by all three models. We locate all pixels that are predicted to have a disparity of less than 40 and depict them in figure 5.4. They would yield a depth of 11.3 cm which according to the CT scan of the colon is impossible. Looking at the output of the softmax function for each of these 192 pixels we can find a pattern. Figure 5.5 illustrates the probability distribution

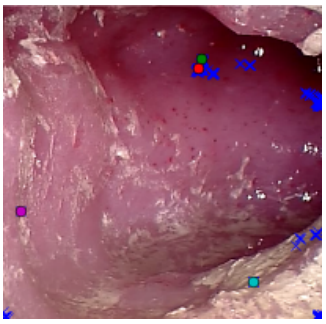


Figure 5.4: Location of incorrectly predicted pixels.

over the disparities between 0 and 128 for some representative examples. Note that the disparity prediction is a classification task, thus the probability distribution is discrete. The two images on top are two pixels (red and green dot in 5.4) from the upper right corner of the image where most errors are found and where a disparity of roughly 60 is expected. It can be observed that the probability distribution of model M2 is very flat and wide. It assigns some positive probability to a majority of pixels. It's certainty is thus quite low. Model M6 predicts several disparities scattered over the entire search space with similar probability but most disparities are assigned zero probability. Intuitively, M6 suggests a few candidates but does not

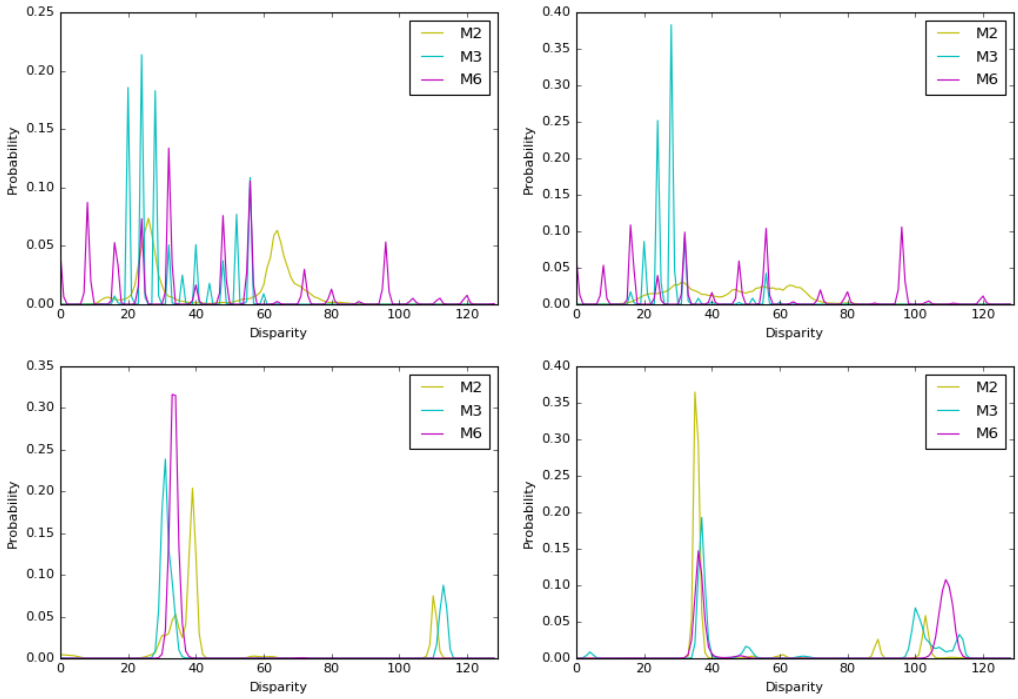


Figure 5.5: Distribution of incorrectly predicted pixels.

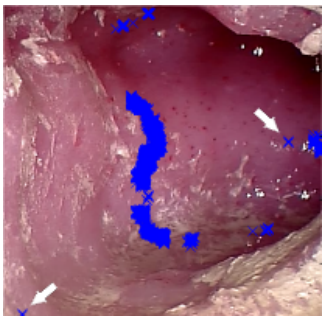


Figure 5.6: Location of correctly predicted pixels.

prefer one explicitly. It assigns high probability to the potentially correct disparity of around 56, even though not the highest which results in a wrong prediction. Model M3 yields a steeper probability function and concentrates on an interval which is a wanted behaviour; however results in a wrong prediction in this case. The lower left image in figure 5.5 corresponds to the blue dot in figure 5.4. The lower right image corresponds to the purple dot. The true disparity of both dots is expected to be > 80. Model M3 fails to find any resemblance between the blue pixel in the left image and the correct area in the right image. Models M2 and M6 both yield a bimodal probability distribution but fail to chose the correct global maximum. We observe the same behaviour for the purple pixel, where all models have two to four local maxima and prefer the incorrect one.

We performed the same procedure for pixels that are correctly predicted by all three models. We identified the 1805 pixels that are predicted to have a disparity of 60 by all three models and made two observations. Firstly, all pixels, with the

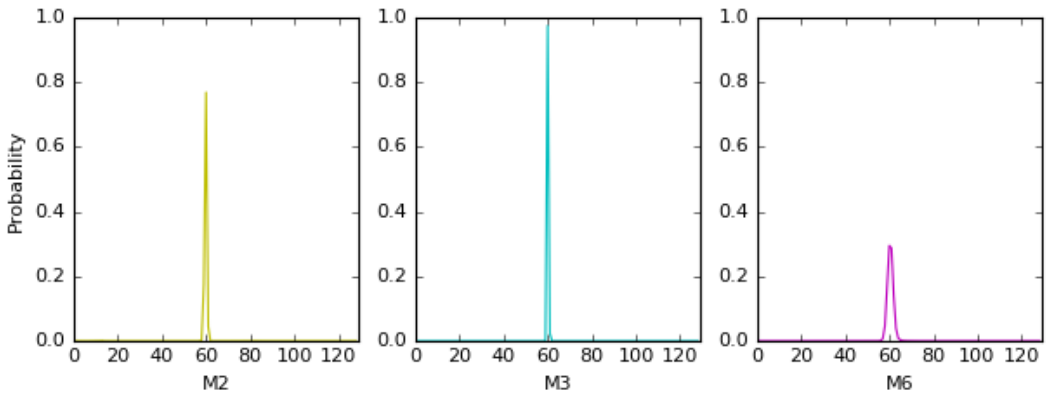
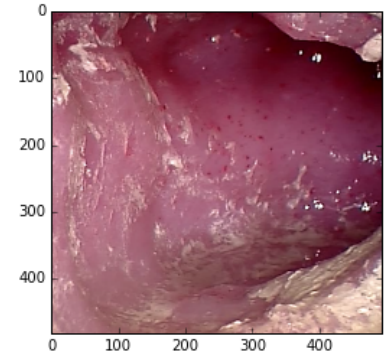
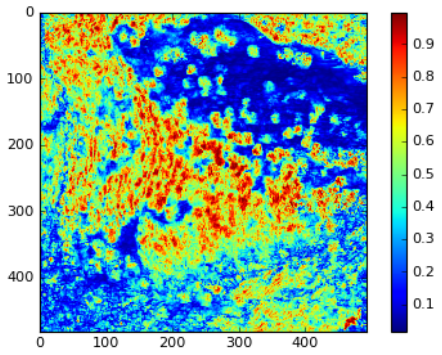


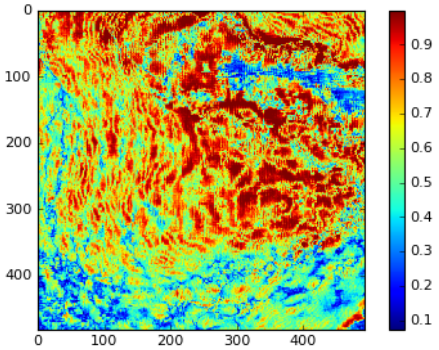
Figure 5.7: Distribution of correctly predicted pixels.



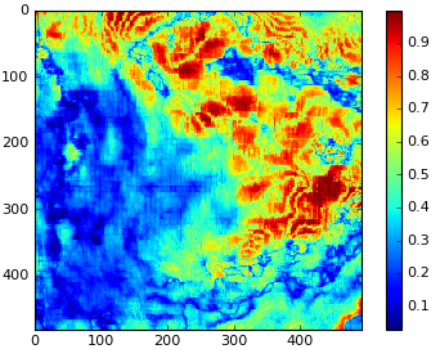
Original image of the phantom.



Probability map of model M2.



Probability map of model M3.

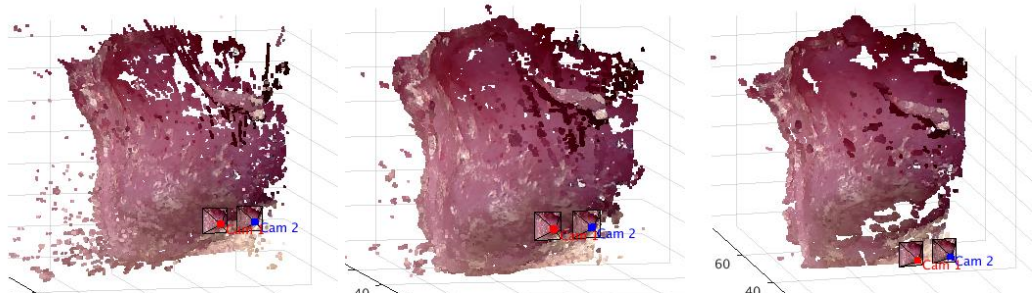


Probability map of model M6.

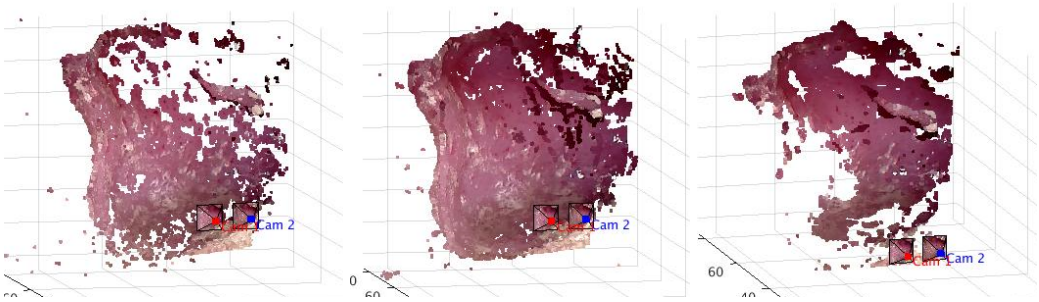
Figure 5.8: Comparison of probability maps of different models.

exception of two, lie perfectly aligned along the shape of the cross section of the colon as illustrated in figure 5.6. This is indicator that the predictions are plausible and likely to be correct. We further depict the probability distributions over the disparities for an arbitrary representative pixel in figure 5.7. All distributions are uni-modal and do not allow any ambiguity, which is another strong indicator for the correctness of the prediction.

Maybe surprisingly model M2 is more than twice as certain as M6. To get to the bottom of this we look at the certainty maps of our three models. These are depicted in figure 5.8. We can observe that each model has a distinct probability map. Model M2 has high certainty for pixels in the middle left part of the image. This is the area where there is more texture and detail. It is less certain in the back of the colon. M3 is in contrast to the other models certain in most parts of the images. The probability map of model M6 is almost the counterpart to that of M2. M6 is



From left to right: M2, M3 and M6 with a threshold of 20%.



From left to right: M2, M3 and M6 with a threshold of 50%.

Depicted are several reconstructions of the same frame with different models and different thresholds. Pixels for which the certainty of the disparity does not exceed this threshold are filtered out.

Figure 5.9: Different reconstructions of the colon.

certain in the upper right, dark area but less so in the detailed area in the lower left of the image.

Before we discuss potential reasons we look at the 3D reconstruction yielded by each one of the models. In figure 5.9 we compare the reconstruction of the three models of the same single frame. After we obtain a disparity map we triangulate each pixel and remove all points in the cloud for which the corresponding output of the softmax function does not exceed a fixed threshold t. The idea is to remove ambiguous predictions like those we saw in figure 5.5. Depicted are the results for $t = 0.2$ and $t = 0.5$. Pixels that are obvious outliers and deviate from the mean 3D coordinates by more than three standard deviations are removed.

It can be observed in figure 5.9 that in accordance with the results on the KITTI dataset model M3 visually yields the best reconstruction. The shape of all six outputs is similar and resembles the CT scan of the colon. The reconstruction with

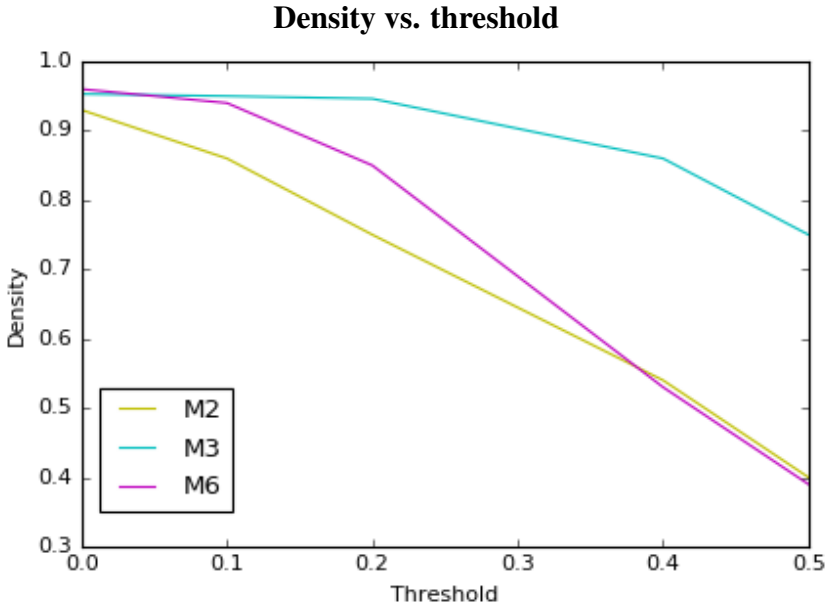


Figure 5.10: Density of predicted pixels after application of different thresholds

models trained on city images yields surprisingly accurate matches, underlining how powerful neural networks are - our models learned the task of matching pixels independently of the scene. However, it can be observed that there is a trade-off between density and accuracy. The lower the threshold the noisier the result but the higher the density of the point cloud. Model M2 (most left images) yields fairly noisy results. The outliers are largely removed when a 50% threshold is applied, however there remain still more outliers than for model M3 (centre images) and for model M6 (most right images). Models M3 and M6 appear to have a fairly similar ratio of outliers to good matches.

We can observe that model M3 yields the most dense reconstruction for any threshold. Model M2 and M6 both lack certainty about the disparity in particular areas of the colon as we have observed before on the probability maps of the models. This observation provides insight into the models' strengths and weaknesses. Model M2 - we recall the seven layer model without pooling layers - is especially weak at the badly illuminated regions in the back of the colon or the right upper corner of the image. This area has less texture than the rest of the colon. Opposed to this, M6, the nine layer model with three pooling layers, yields especially poor results in the front of the phantom, the lower left corner of the image. This area is very detailed due

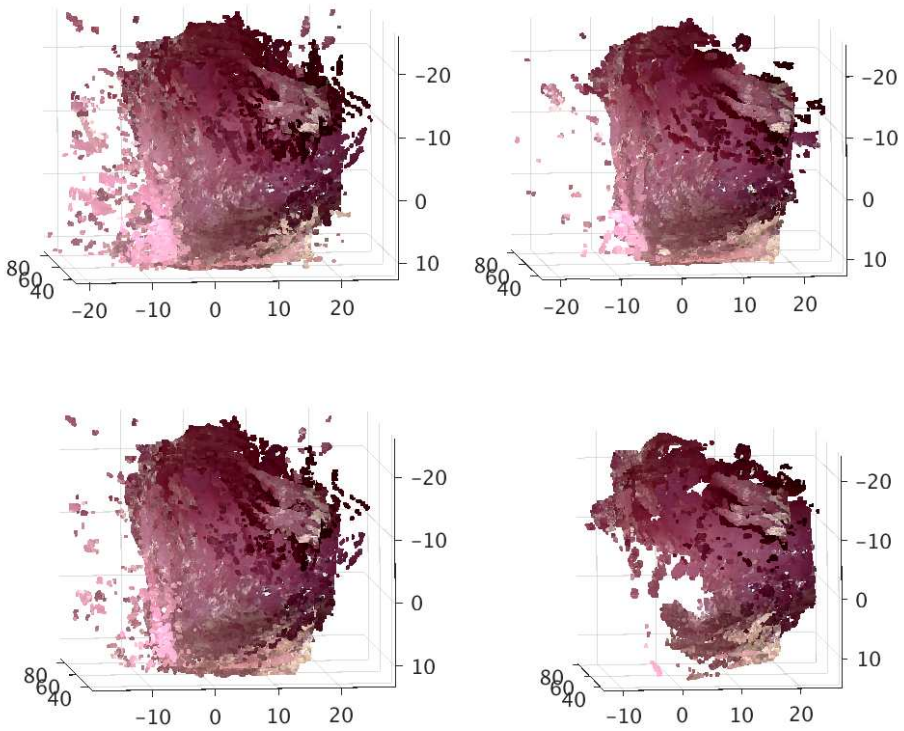
to the imitated sediments of the remaining liquids in the colon. These observations coincide with the weaknesses and strengths of pooling layers: they incorporate a wider receptive field, therefore model M6 predict the areas with little texture better than model M2. However, they blur details, which is why model M2 outperforms model M6 on areas with a lot of details. M3 seems to be a good compromise.

To quantify the results we plot the density $dens(t)$ of a model against its certainty as a measure of its potential to predict disparity. Calculating the densities for models M2, M3 and M6, and for thresholds 0, 0.1, 0.2, 0.4 and 0.5 we obtain the graph in figure 5.10.

We can observe clearly that the density of model M3 is the highest for all thresholds but zero. The difference becomes more apparent the higher the threshold is. The density is not a measure for the performance of the models. However, taking into account the noise vs. density pattern we observed in figure 5.9 we can draw some conclusions. In particular, we saw that applying a threshold of 0.5 removes almost all outliers for models M3 and M6. The density we obtain for a threshold of 0.5 therefore must be close to the ratio of correctly matched pixels to all pixels, which indeed is a measure of performance. Observing the high discrepancy between their densities - 75% vs. 39% out of the 237,133 pixels - we can argue that model M3 yields the best results on the reconstruction of the colon.

So far we have only looked at results obtained from one frame. We now discard model M2 and use an iterative closest point (ICP) algorithm to align the point clouds from several frames. We apply MATLAB's *pcregrigid* function according to [92] to register triangulated points from the four frames depicted in figure 4.10. Observing figure 5.11 we make the same observations as before: M3 yields a higher density but also more obvious errors than M6. Observing the inner surface of the reconstruction it is visible that model M3 preserves more detail, which is why based on this qualitative analysis we conclude that model M3 is better suited to reconstruct the inner surface of the phantom.

After this qualitative discussion we conduct a quantitative evaluation. Computer aided diagnosis and surgery requires an exact localisation in space. Misjudg-



Top: Threshold of 0.2. Bottom: Threshold of 0.5.
Left: Model M3. Right: Model M6.

Figure 5.11: 3D model of the phantom of a colon reconstructed from four frames.

ing distances by a couple of millimetres could cause severe injuries. Therefore we triangulate the stereo correspondences and compare the obtained 3D model directly to the CT scan of the phantom, rather than comparing ratios of correctly estimated disparities. To compare the accuracy of the models, we used *CloudCompare*, a software that compares point clouds or mashes. A caveat of this approach is that several steps along both, the reconstruction pipeline of our models and the comparison itself, cause noise. The results of this approach therefore gives a reasonable range of the matching error but should be interpreted with caution. Errors in the registration that are not caused by the matching can result from:

- Calibration of the cameras
- Rectification of the images

- ICP algorithm
- Manual initial alignment of the two point clouds
- Segmentation of the CT

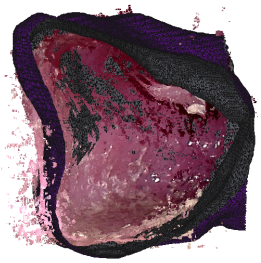
In our case the manual adjustment as starting point for the fine alignment is based on visual judgement of the correct position. Ideally, we would obtained the CT model of the phantom with the camera inserted. This would enable a mapping between the frames of the video and the camera position relative to the phantom.

The ICP algorithm used for the alignment of the point cloud of our predicted colon model to the CT mash minimises the RMS, the square root of the mean squared error between each point of the cloud and its nearest point on the mash. To this end, the two point clouds first have to be roughly aligned by hand. After the registration the mean distance between the cloud and the mash can be calculated as average euclidean distance between each point and its closest point on the mash. It has to be noted, that the nearest point on a mash is not necessarily the correct point, therefore the mean distance should not be used as the only measure of accuracy.

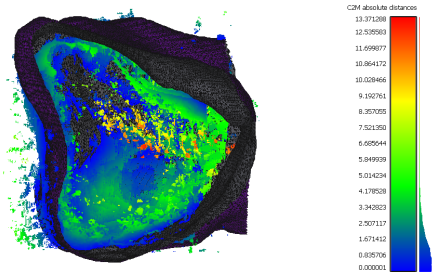
Model	Threshold	Mean distance*	RMS*	points in cloud
M3	0.2	1.63	2.27	**796,458**
M3	0.5	1.46	2.24	599,388
M6	0.2	1.50	2.03	691,102
M6	0.5	1.41	1.92	308,403
[15]	-	**1.17**	1.50	529,735

Table 5.2: Reconstruction error. * in mm

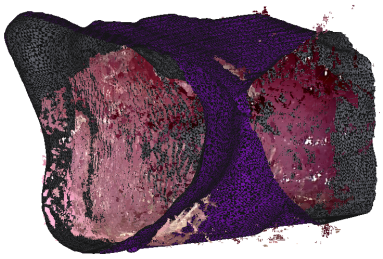
In table 5.2 we compare models M3 and M6 with two different thresholds. M6 with a threshold of 0.5 yields as expected the smallest reconstruction error but also the smallest number of matches. M3 with a threshold of 0.2 maintains the largest number of matched points exceeding the threshold. Because M3 yields with a mean distance of 1.46 mm only a slightly worse result than model M6 with 1.41, but has an almost two times higher density, we propose model M3 as our best model. However, the comparison of the models is in general limited, because we do not necessarily compare the same points. The threshold can be maintained as parameter and



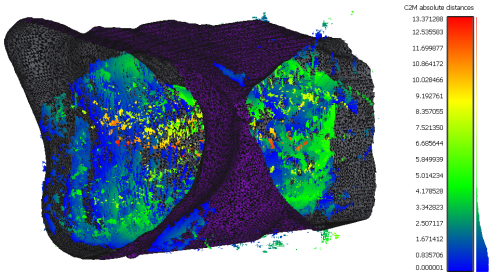
Front view on cloud and ground truth.



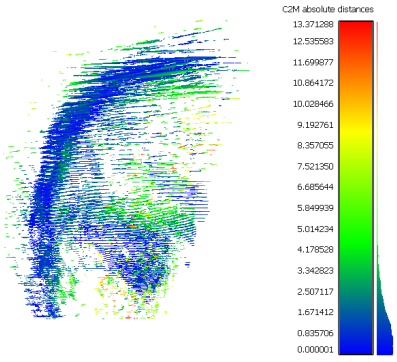
Front view on errors and ground truth.



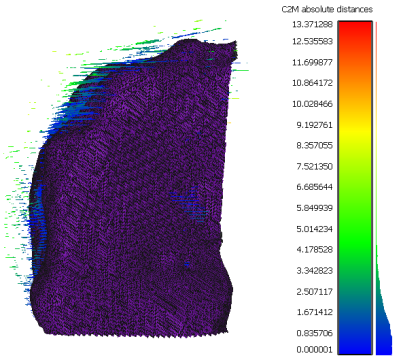
Side view on cloud and ground truth.



Side view on errors and ground truth.



Top view on errors.

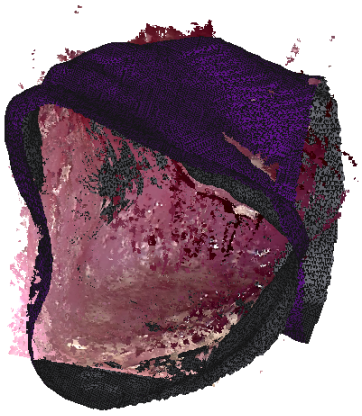


Top view on errors and ground truth.

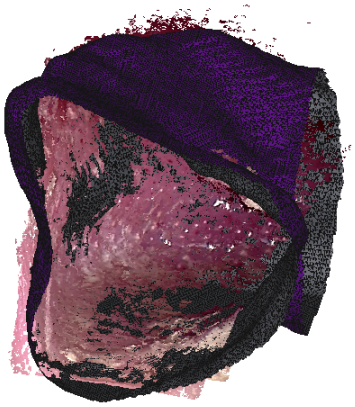
Figure 5.12: Comparison of the predicted 3D model and the ground truth

set according to the dataset. While for us usually around 0.4-0.5 is reasonable, an implementation of a proper post-processing pipeline could enable a lower threshold. The aligned point cloud of model M3 is shown in figure 5.12.

Lastly, we applied the stereo matching algorithm of Stoyanov *et al.* [15] to the same four frames of the colon. The algorithm yields a mean distance of 1.17 mm from the CT and matches roughly 530k points. The different reconstructions can be seen in figure 5.13. Although our model yields a slightly higher error on this test set, our reconstruction matches more points (roughly 600k) and covers a larger area



Front view on ground truth and reconstructed model using model M3.



Front view on ground truth and reconstructed model using [15].

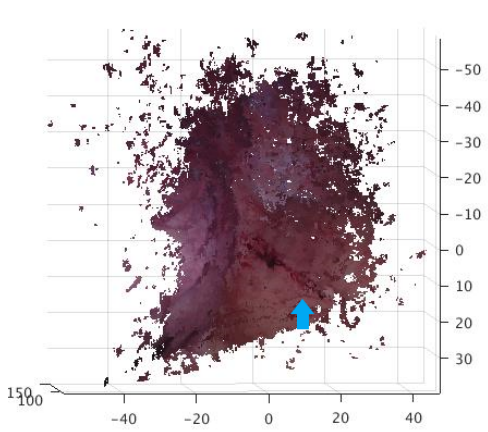
Top view on reconstructed model using model M3.

Top view on reconstructed model using [15].

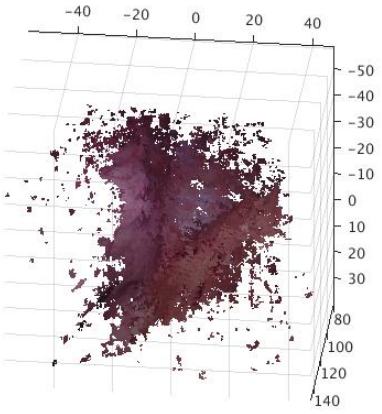
Figure 5.13: Comparison of our best model and the state-of-the-art.

of the ground truth, yielding a more dense reconstruction. However, noise could result from the registration and is not necessarily result of the different matching algorithms. We therefore argue that our model yields a comparable mean distance, but a more dense reconstruction than established stereo matching algorithms for robotically assisted minimally invasive surgery.

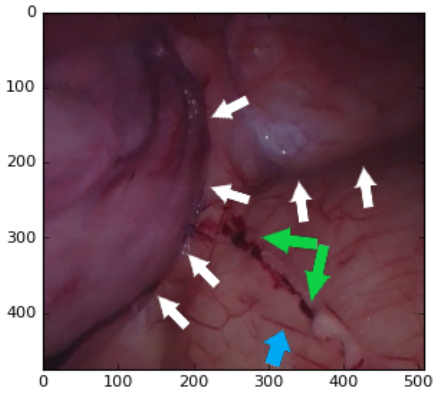
5.4 Reconstruction of Further Intraoperative Scenes



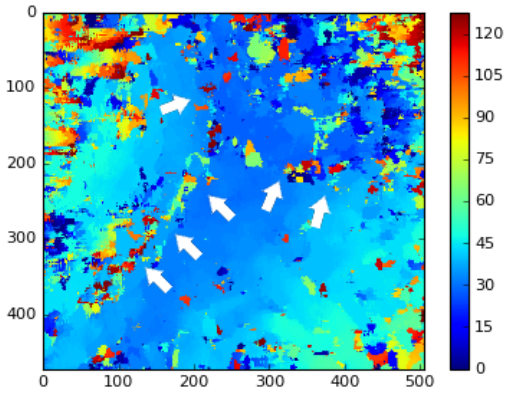
Point cloud of scene: View from left.



Point cloud of scene: View from lower left.



Original image



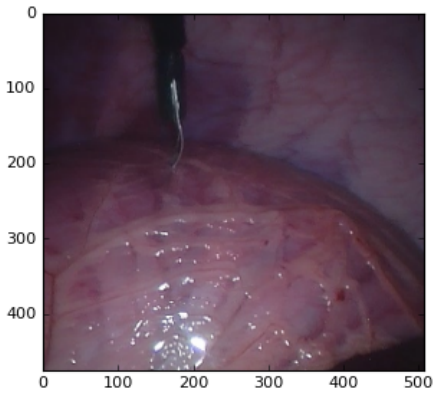
Disparity map

Figure 5.14: Example 1: Evaluation on porcine test set.

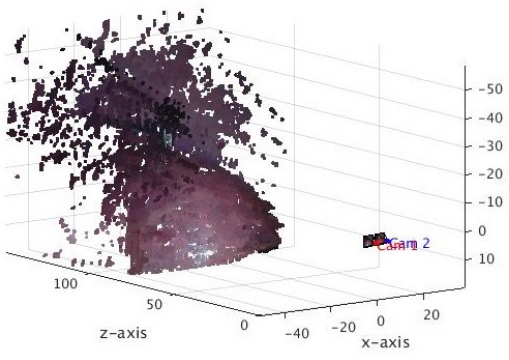
As extension to the project we applied our algorithm to images of real tissue in order to investigate the performance on real data. The present dataset contains of two videos of stereoscopic images of porcine intestines. The tissue is similar to the inner surface of the colon and the quality of the images regarding resolution, lightening

and frame rate. However, the shape of the scene is not tubular. We applied models M3 and M6 just as we did for the city and colon images. Results of the first video are depicted in figure 5.14. We found that model M6 did not correctly reconstruct the large blood vessel in the lower right corner of the image indicated by green arrows. Model M3 yields a more accurate point cloud and even restores the very fine blood vessels like the one indicated with a blue arrow in the original image and the point cloud. Exploring the matches before the triangulation we observed that there are more obvious mismatches than in the colon dataset. We trace this to the considerably finer details, the smoother surface, and the worse illumination within the abdominal cavity. Further, the occluded areas in the folds of the tissue (indicated by white arrows in figure 5.14) yield considerable mismatches due to the geometry of the observed scene. In the disparity map in figure 5.14 the disparities along the folds are partially larger than 75 but should be smaller than the surrounding tissue, because greater depth corresponds to smaller disparity. The deep folds can only be observed from a very narrow angle. The two cameras whose optical centres are around 5 mm apart of each other receive contradictory results. Especially in these areas, where the geometry of a scene prevents stereo view, smoothing as post-processing step is indispensable. Nonetheless, there is an impressive amount of detail in the reconstruction.

In figure 5.15 we reconstructed a scene that was interesting because it shows



Original image



3D model side view

Figure 5.15: Example 2: Evaluation on porcine test set.

surgical instruments. Because the instruments were moving in the video, we only report results from a single frame. The view from left indicates that the shape is correctly reconstructed. There is a high level of accuracy in the image. The upper left corner causes a considerable amount of noise, however even the position of the instrument is obtained, although it has barely any texture.

Chapter 6

Conclusions and Future Work

In this thesis we implemented a dense stereo reconstruction algorithm to obtain a 3D model of the colon that has the potential to improve colorectal cancer diagnosis. As opposed to conventional stereo matching algorithms for surgical applications, we approached the problem using Convolutional Neural Nets to learn a similarity measure from data instead of imposing a hand-crafted cost function. We focused on finding a network architecture that is best suited for the reconstruction of intraoperative environments, which typically have little texture and few edges. The key idea was to increase the size of the receptive field of the model to enable the incorporation of more context and therefore a more accurate prediction in regions without texture. We compared different models and observed the impact of the size of the receptive field on the accuracy of the reconstructed scene. We found that our architectures have the potential to outperform models without pooling-layers considerably by

- down-sampling the feature representation of the images using pooling layers and
- up-sampling the encoded feature representation of the images afterwards with transposed convolutions in order to maintain the initial image resolution

This approach forces the model to include information from a wider range of neighbouring pixels and, not only yields a higher accuracy, but also a smoother result. We applied our Siamese CNN to stereo images of a colon phantom and validated

our results by comparing our 3D model to a computed tomography (CT) scan of the phantom. Our best architecture yields a mean distance between our predicted 3D point cloud and the CT of 1.46 mm. We showed that our model yields a denser reconstruction than established stereo matching algorithm for minimally invasive surgery and a comparable mean distance to the CT. We see a lot of potential in our approach to outperform conventional stereo matching algorithms on colonoscopy images considerably, once post-processing is applied. Luo *et al.* showed for instance that solely post-processing can result in a three times smaller 3 pixel error [83]. Moreover, we showed that our model is also applicable to both, *in vivo* scenes, and images of traffic, cities, and cars. This application under very different conditions, shows that our model learned a universal similarity measure.

When comparing two models with receptive fields of 15 x 15 or 56 x 56 pixels, respectively, we found that the model with the small receptive field outperforms the second considerably on scenes with much detail. On the other hand, the model with a large receptive field performs a lot better on scenes without texture. Future work will therefore focus on an architecture where two receptive fields are implemented parallely and the resulting feature representation concatenated afterwards. This way a model could take advantage of the best of both variants and balance the trade-off between an increased receptive field and loss of detail. Moreover, we will implement post-processing steps, which could possibly be done by incorporating additional layers in the network.

Given the scale of the colon (diameter of roughly 2.5 cm) we believe that this technology has he potential to result in a very accurate reconstruction of the colon that will improve the detection rate of colorectal cancer.

Bibliography

[1] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4):198–211, 2007.

[2] Lindsey A Torre, Freddie Bray, Rebecca L Siegel, Jacques Ferlay, Joannie Lortet-Tieulent, and Ahmedin Jemal. Global cancer statistics, 2012. *CA: a cancer journal for clinicians*, 65(2):87–108, 2015.

[3] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, Olivier Romain, Bjørn Rustad, Ilangko Balasingham, et al. Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai 2015 endoscopic vision challenge. *IEEE Transactions on Medical Imaging*, 36(6):1231–1249, 2017.

[4] Colorectal cancer. *Encyclopedia Britannica*, 2017.

[5] Gina Brown. *Colorectal Cancer*. Cambridge University Press, 2007.

[6] Jerome D Waye, Douglas K Rex, and Christopher Beverley Williams. *Colonoscopy: principles and practice*. Wiley Online Library, 2009.

[7] John H Scholefield and Cathy Eng. *Colorectal Cancer: Diagnosis and Clinical Management*. John Wiley & Sons, 2014.

[8] Jessica B OConnell, Melinda A Maggard, and Clifford Y Ko. Colon cancer survival rates with the new american joint committee on cancer sixth edition staging. *Journal of the National Cancer Institute*, 96(19):1420–1425, 2004.

[9] Michal F Kaminski, Jaroslaw Regula, Ewa Kraszewska, Marcin Polkowski, Urszula Wojciechowska, Joanna Didkowska, Maria Zwierko, Maciej Rupinski, Marek P Nowacki, and Eugeniusz Butruk. Quality indicators for colonoscopy and the risk of interval cancer. *New England Journal of Medicine*, 362(19):1795–1803, 2010.

[10] Perry J Pickhardt, Pamela A Nugent, Pauline A Mysliwiec, J Richard Choi, and William R Schindler. Location of adenomas missed by optical colonoscopy. *Annals of internal medicine*, 141(5):352–359, 2004.

[11] JEG IJspeert, CJ Tutein Nolthenius, EJ Kuipers, ME Van Leerdam, CY Nio, MGJ Thomeer, Katharina Biermann, MJ Van De Vijver, Evelien Dekker, and Jacob Stoker. Ct-colonography vs. colonoscopy for detection of high-risk sessile serrated polyps. *The American journal of gastroenterology*, 111(4):516–522, 2016.

[12] Peter B Cotton, Valerie L Durkalski, Benoit C Pineau, Yuko Y Palesch, Patrick D Mauldin, Brenda Hoffman, David J Vining, William C Small, John Affronti, Douglas Rex, et al. Computed tomographic colonography (virtual colonoscopy): a multicenter comparison with standard colonoscopy for detection of colorectal neoplasia. *Jama*, 291(14):1713–1719, 2004.

[13] Dan Koppel, Chao-I Chen, Yuan-Fang Wang, Hua Lee, Jia Gu, Allen Poirson, and Rolf Wolters. Toward automated model building from video in computer-assisted diagnoses in colonoscopy. In *Proceedings of the SPIE Medical Imaging Conference*, pages 1117–1120. San Diego, CA, Bellingham, USA: SPIE, 2007.

[14] Nicholas J Durr, Germán González, and Vicente Parot. 3d imaging techniques for improved colonoscopy, 2014.

[15] Danail Stoyanov, Marco Scarzanella, Philip Pratt, and Guang-Zhong Yang. Real-time stereo reconstruction in robotically assisted minimally invasive

surgery. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, pages 275–282, 2010.

[16] Lena Maier-Hein, Anja Groch, Adrien Bartoli, Sebastian Bodenstedt, G Boissonnat, P-L Chang, NT Clancy, Daniel S Elson, Sven Haase, Eric Heim, et al. Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction. *IEEE transactions on medical imaging*, 33(10):1913–1930, 2014.

[17] Lazaros Nalpantidis and Antonios Gasteratos. Biologically and psychophysically inspired adaptive support weights algorithm for stereo correspondence. *Robotics and Autonomous Systems*, 58(5):457–464, 2010.

[18] Sebastian Röhl, Sebastian Bodenstedt, Stefan Suwelack, Hannes Kenngott, Beat P Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Dense gpu-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration. *Medical physics*, 39(3):1632–1645, 2012.

[19] Ping-Lin Chang, Danail Stoyanov, Andrew J Davison, et al. Real-time dense stereo reconstruction using convex optimisation with a cost-volume for image-guided robotic surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 42–49. Springer, 2013.

[20] Evaggelos Spyrou, Dimitris Diamantis, and Dimitris K Iakovidis. Panoramic visual summaries for efficient reading of capsule endoscopy videos. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2013 8th International Workshop on*, pages 41–46. IEEE, 2013.

[21] Qian Zhao and Max Q-H Meng. 3d reconstruction of gi tract texture surface using capsule endoscopy images. In *Automation and Logistics (ICAL), 2012 IEEE International Conference on*, pages 277–282. IEEE, 2012.

[22] Gastone Ciuti, Marco Visentini-Scarzanella, Alessio Dore, Arianna Menciassi, Paolo Dario, and Guang-Zhong Yang. Intra-operative monocular 3d

reconstruction for image-guided navigation in active locomotion capsule endoscopy. In *Biomedical Robotics And Biomechatronics (Biorob), 2012 4th Ieee Ras & Embs International Conference On*, pages 768–774. IEEE, 2012.

[23] Arie Kaufman and Jianning Wang. 3d surface reconstruction from endoscopic videos. *Visualization in Medicine and Life Sciences*, pages 61–74, 2008.

[24] Chao-I Chen, Dusty Sargent, and Yuan-Fang Wang. Modeling tumor/polyp/lesion structure in 3d for computer-aided diagnosis in colonoscopy. In *Proc. of SPIE Vol*, volume 7625, pages 76252F–1, 2010.

[25] DongHo Hong, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C De Groen. 3d reconstruction of colon segments from colonoscopy images. In *Bioinformatics and BioEngineering, 2009. BIBE'09. Ninth IEEE International Conference on*, pages 53–60. IEEE, 2009.

[26] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[27] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.

[28] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 3d shape scanning with a time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1173–1180. IEEE, 2010.

[29] David Marr and Tomaso Poggio. Cooperative computation of stereo disparity. In *From the Retina to the Neocortex*, pages 239–243. Springer, 1976.

[30] Simon JD Prince. *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[31] Depth map from stereo images. `http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_depthmap/py_depthmap.html`. Accessed: 2017-09-02.

[32] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.

[33] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[34] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[35] Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. *Computer VisionECCV 2002*, pages 8–40, 2002.

[36] Sébastien Roy and Ingemar J Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Computer Vision, 1998. Sixth International Conference on*, pages 492–499. IEEE, 1998.

[37] Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng. Stereo matching using belief propagation. In *European Conference on Computer Vision*, pages 510–524. Springer, 2002.

[38] Heiko Hirschmüller, Peter R Innocent, and Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, 2002.

[39] Yuan C Hsieh, David M McKeown Jr, and Frederic P Perlant. Performance evaluation of scene registration and stereo matching for artographic feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):214–238, 1992.

[40] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[41] Marsha J Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

[42] BR Hunt. Prediction of correlation errors in stereo-pair images. *Optical Engineering*, 19(3):31–312, 1980.

[43] Daniel Scharstein. Matching images by comparing their gradient fields. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 572–575. IEEE, 1994.

[44] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European conference on computer vision*, pages 151–158. Springer, 1994.

[45] Minglun Gong, Ruigang Yang, Liang Wang, and Mingwei Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *International Journal of computer vision*, 75(2):283–296, 2007.

[46] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 807–814. IEEE, 2005.

[47] Andrea Fusiello, Vito Roberto, and Emanuele Trucco. Efficient stereo with multiple windowing. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 858–863. IEEE, 1997.

[48] Takeo Kanade and Masatoshi Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE transactions on pattern analysis and machine intelligence*, 16(9):920–932, 1994.

[49] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A com-

parative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080, 2008.

[50] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[51] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003.

[52] Martin J Wainwright, Tommi S Jaakkola, and Alan S Willsky. Map estimation via agreement on trees: message-passing and linear programming. *IEEE transactions on information theory*, 51(11):3697–3717, 2005.

[53] Vladimir Kolmogorov and Carsten Rother. Comparison of energy minimization algorithms for highly connected graphs. In *Proceedings of the 9th European conference on Computer Vision-Volume Part II*, pages 1–15. Springer-Verlag, 2006.

[54] Qi Tian and Michael N Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35(2):220–233, 1986.

[55] Li Cheng and Terry Caelli. Bayesian stereo matching. *Computer Vision and Image Understanding*, 106(1):85–96, 2007.

[56] Li Zhang and Steven M Seitz. Estimating optimal parameters for mrf stereo from a single image pair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2), 2007.

[57] Vladimir Kolmogorov, Antonio Criminisi, Andrew Blake, Geoffrey Cross, and Carsten Rother. Probabilistic fusion of stereo with color and contrast for bilayer segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1480–1492, 2006.

[58] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[59] Yunpeng Li and Daniel P Huttenlocher. Learning for stereo vision using the structured support vector machine. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[60] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[61] Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

[62] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4):391–444, 2007.

[63] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[64] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[65] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[67] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[68] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[69] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[70] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[71] David Barber. Neural nets, April 2017.

[72] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989.

[73] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[74] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[75] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[76] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[77] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[78] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[79] V Dumoulin. Convolution arithmetic. `https://github.com/vdumoulin/conv_arithmetic`. Accessed: 2017-08-31.

[80] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[81] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1592–1599, 2015.

[82] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32):2, 2016.

[83] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5695–5703, 2016.

[84] Haesol Park and Kyoung Mu Lee. Look wider to match image patches with convolutional neural networks. *IEEE Signal Processing Letters*, 2016.

[85] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[86] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[87] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. *arXiv preprint arXiv:1703.04309*, 2017.

[88] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[90] Amit Shaked and Lior Wolf. Improved stereo matching with constant highway networks and reflective confidence learning. *arXiv preprint arXiv:1701.00165*, 2016.

[91] Rivo Öpik, Andres Hunt, Asko Ristolainen, Patrick M Aubin, and Maarja Krusmaa. Development of high fidelity liver and kidney phantom organs for use with robotic surgical systems. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 425–430. IEEE, 2012.

[92] Yang Chen and Gérard Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724–2729. IEEE, 1991.