Stefan HEBER

3D Image Reconstruction Using Active Wavefront Sampling

DIPLOMARBEIT

zur Erlangung des akademischen Grades einer/s Diplom-Ingenieur/in

Diplomstudium Technische Mathematik

Graz University of Technology

Technische Universität Graz

Betreuer/in:
Univ.-Prof. Dipl.-Ing. Dr.techn. Horst BISCHOF
Dipl.-Ing. Dr.techn. Matthias RÜTHER

Institute for Computer Graphics and Vision (ICG)

Graz, im Juli 2010

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quotes either literally or by content from the used sources.

....
date (signature)

Abstract

Active Wavefront Sampling (AWS) is a 3D surface imaging technique, which uses only a single camera and an AWS module. In its simplest form, an AWS module is an off-axis aperture, that moves on a circular path around the optical axis. By moving the aperture around the optical axis, target points on the image plane rotate on a circle (assuming that we have ideal non-aberrated conditions). The target points depth information is coded by the diameter of the according image rotation. In principle, AWS imaging allows any system with a digital camera to function in 3D. Thus it eliminates the need for multiple cameras to acquire 3D.
In this work we present a global optical flow approach to calculate the blur-circle-radii generated by a rotating AWS module. Thus, we add prior knowledge about the sought depth maps and assume that the blur-circle-radius varies smoothly almost everywhere in the image. Our approach is based on total variation (TV) in the regularization-term and the robust L^1-norm in the data-term. Therefore, the approach is referred to as TV-L^1 global AWS.

Keywords: Active Wavefront Sampling, 3D Reconstruction, global AWS

Kurzfassung

Active Wavefront Sampling (AWS) ist ein Verfahren zur 3D Rekonstruktion von Oberflächen, welches lediglich eine einzelne Kamera mit einem speziellen AWS-Modul benötigt. In der einfachsten Form handelt es sich bei einem AWS-Modul um eine off-axis Blende, welche um die optische Achse rotiert wird. Durch diese Rotation der Blende werden die Bildpunkte ebenfalls in Rotation versetzt. Die Analyse der Rotationsbewegung eines bestimmten Punktes ermöglicht die Berechnung der dazugehörigen Tiefeninformation. Im Prinzip ermöglicht ein AWS-Modul 3D Informationen mit einer beliebigen Kamera zu berechnen, und macht somit ein multi-Kamera-Setup unnötig.
In der vorliegenden Diplomarbeit wird ein Verfahren vorgestellt, das es ermöglicht die Rotationsbewegungen, welche durch ein rotierendes AWS-Modul erzeugt werden, global zu berechnen. Hierfür wird a priori Information über die Oberflächenbeschaffenheit des Objektes für die Berechnung verwendet. Das Verfahren basiert im Wesentlichen auf Total Variation (TV) und der robusten L^1-Norm und wird somit als TV-L^1 global AWS bezeichnet.

Stichwörter: Active Wavefront Sampling, 3D Rekonstruktion, global AWS

Acknowledgements

First of all, I would like to thank Dipl.-Ing Dr.techn. Matthias Rüther for guidance and encouraging engagement in advising. Moreover, I want to thank my thesis supervisor Prof. Dipl.-Ing Dr.techn. Horst Bischof. I would also like to thank my brother Dipl.-Ing Markus Heber for interesting discussions and support. Last but not least, I am deeply grateful for my parents for giving me the opportunity to do my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

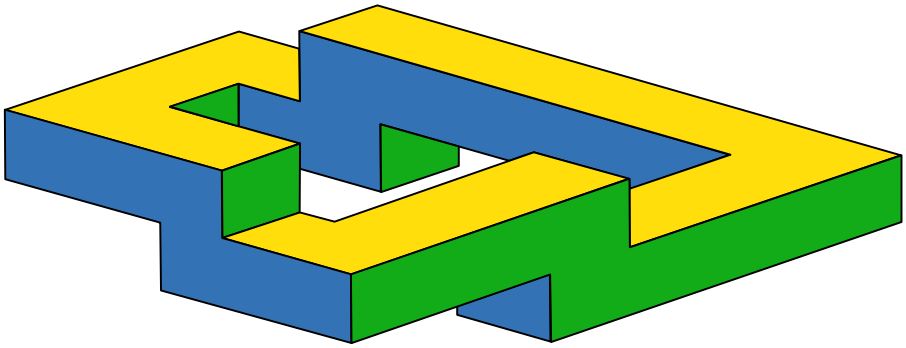


Figure 1.1: Endless stairs illusion. The figure shows an impossible object in which the stairs make four 90° turns as they ascend or descent, which is clearly impossible in 3D. The 2D figure achieves this optical illusion by distorting perspective.

We live in a 3D world and thanks to our eyes we can possess stereo vision. The process in visual perception of capturing depth from slightly different projections of the world onto the retinas of the two eyes is called stereopsis[1]. Depth information is obtained by the difference in the two images called horizontal disparity or binocular disparity. Inspired by this, a fundamental problem in computer vision is to obtain 3D geometric information from captured planar images of an observed scene. This process is traditionally referred to as 3D reconstruction. Due to the fact that these images are 2D projections of our 3D world, all quantitative depth information is lost. Only shading and a priori knowledge of the scene allow us to qualitatively extract depth cues from such 2D images. But also these qualitative depth cues can fool a human observer. Consider for example depth ambiguities also known as optical illusions (cf Fig. 1.1).
A transition from 2D to 3D image capture and display is extremely desirable. It is in fact hard to imagine an imaging application, where the added capability to measure depth would not be welcome. Therefore, the acquired 3D models have a large field of application, including 3D

[1] from stereo meaning solidity, and opsis meaning vision or sight

model capture for the movie and entertainment industries, computer aided design, industrial design, reverse engineering, prototyping, and quality control. However, if 3D scanners should become more ordinary, new methods, that are able to acquire robust geometric models of real objects fast and easily using low cost technologies, are needed. When these conditions are once complied, then 3D imaging will replace 2D imaging, due to the simple fact, that the world is not flat.

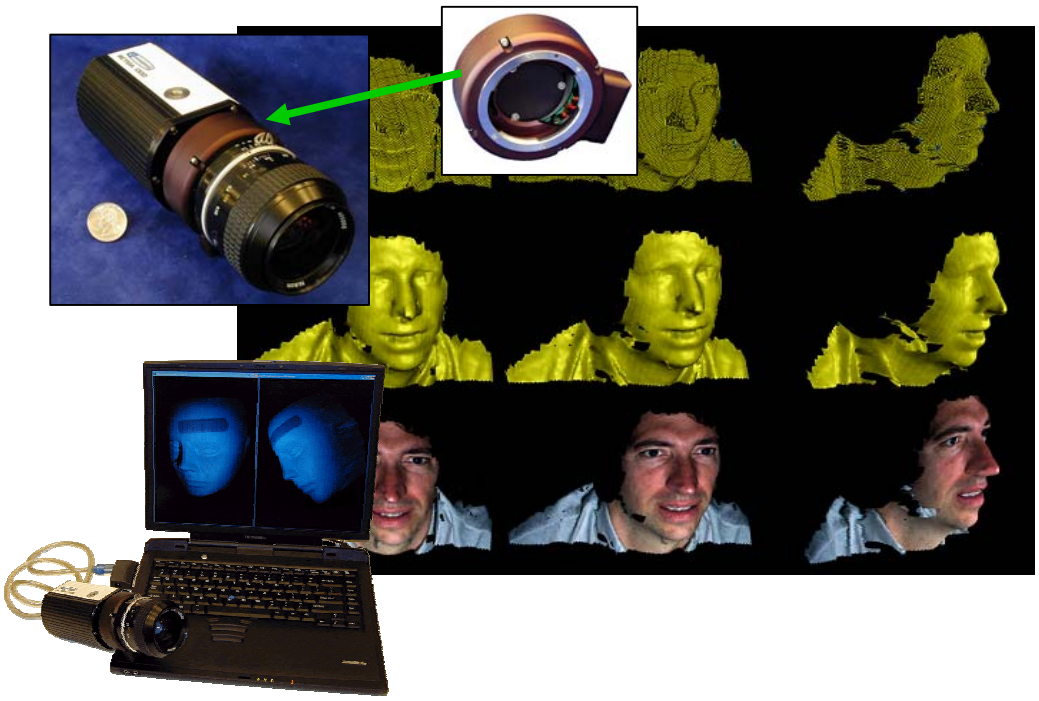


Figure 1.2: MIT's Active Wavefront Sampling (AWS) 3D imaging system. The image shows a CMOS USB camera with a Nikon 35mm lens and an additional AWS module. The images are processed and the 3D models are acquired at video rates. [11]

The Active Wavefront Sampling (AWS) approach presented by Frigerio [10] makes a start on lowering the system costs for a 3D scanner, as well as providing high accurate 3D models (cf Fig. 1.2). Additionally it is a passive vision system with a simple hardware, which allows a very compact setup. The method requires only a single camera with a special rotating off-axis aperture to capture depth information. Thus, in principle it allows any system with a digital camera to function in 3D. Moreover, it halves the costs for the optics compared to a standard stereo setup. Furthermore, the method can be easily adjusted to the given requirements, meaning that it allows realtime reconstruction, by using a low number of aperture positions, as well as high precise reconstruction, by using a high number of aperture positions. The AWS system is also quite small and needs to be calibrated only once, contrary to a stereo setup that needs to be calibrated whenever the relative position between the cameras is modified. Although the AWS technique can be used to reconstruct any kind of object, it works best for objects, that are very close to the camera. Therefore, typical

applicational examples of AWS include minimally invasive surgery using an endoscope, and 3D microscopy.

Due to the above mentioned advantages of the AWS system it is reviewed in this thesis. Beside reviewing the suggested methods from Frigerio, we, moreover, present a global optical flow method to solve the AWS problem, where we assume that the blur-circle-radius varies smoothly almost everywhere in the image. Therefore, this thesis concentrates on the problem of extracting 3D structures by analyzing the motion in visual scenes.
In order to improve the accuracy of the calculated 3D models, acquired by the algorithms presented by Frigerio, we concentrate our research on the image motion obtained by moving the off-axis aperture of an AWS system. We introduce a special motion model to calculate the optical flow field obtained by an AWS system. Thus, we incorporate the circle approximation into the optical flow calculation, which are separated steps in the Frigerio multi image approach. Moreover we add prior knowledge about the sought depth maps and develop a global approach to solve the according AWS minimization problem.
By testing the approach on synthetic image sequences we show that our approach reaches slightly better accuracy results than the Frigerio version and it is also robust in the presence of Gaussian image noise. Moreover our global AWS approach is capable of parallel processing and can hence be accelerated on the GPU.

The thesis is organized as follows:

Chapter 2 provides an overview of current 3D reconstruction techniques, including the wavefront sampling approach. Here the BIRIS sensor, as an example for Static Wavefront Sampling (SWS), and the main idea of the Active Wavefront Sampling (AWS) approach will be presented.

Next, Chapter 3 presents necessary background information to clearly understand the AWS approach. First, a brief review on visual motion, and optical flow is presented, followed by a survey of basic concepts about the theory of defocus and wavefront sampling.

Chapter 4 present a new global AWS approach, which assumes that the blur-circle-diameter varies smoothly almost everywhere in the image. For a better understanding, first a version of the global AWS approach is presented, that uses the L^2-norm to weight the data-term. In a second step, a robust version of the global AWS approach is presented, that uses the L^1-norm to weight the data-term, as well as an edge-preserving smoothness-term.

Finally, in Chapter 5 we evaluate the different AWS approaches and present some experimental results as well as 3D reconstruction results.

Chapter 2

Related Work

Contents

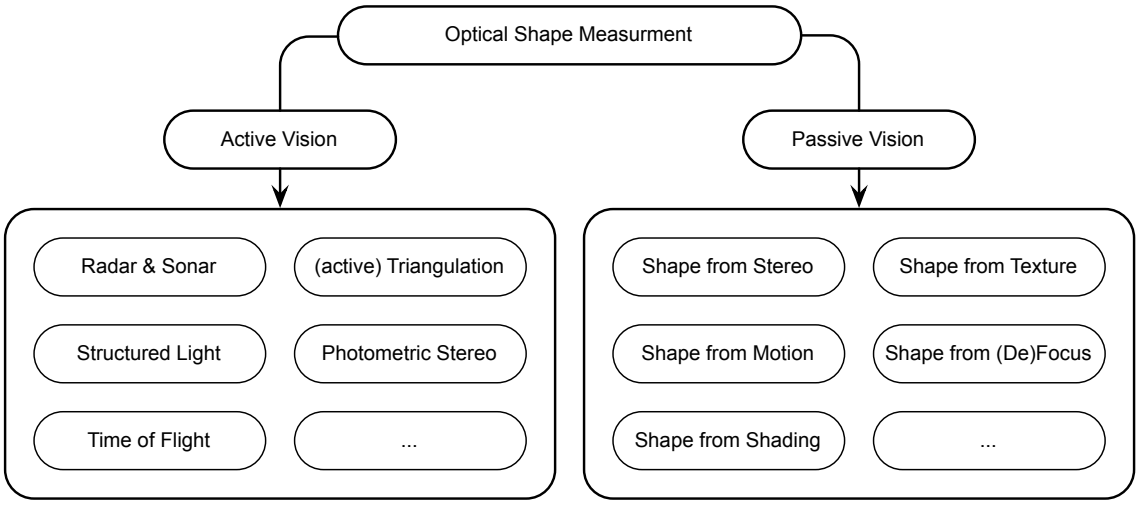


Figure 2.1: Classification of non contact optical shape measurement techniques.

There are generally two types of (non-contact) 3D reconstruction techniques called active vision and passive vision (cf Fig. 2.1).

Active vision techniques use a controlled source of structured energy emission, e.g. some kind of radiation or light, and an additional detector, e.g. a camera. The 3D data is recovered by using the nature of the interaction of the reflected source with the object surface. However, active vision methods have some drawbacks. It may be difficult to use them in outside applications, or more general, when there are conflicting ambient energy sources. Furthermore, variation in object reflectance and color may have a negative influence on the accuracy.
On the other hand, passive vision techniques do not use a specific structured source of energy. The basic principle used in many passive vision methods is the so called triangulation principle. This principle refers to the idea that when the distance between two points and the angles from each of these points to a third point are known, then the location of the third point can be calculated. In triangulation-based methods one of the two points must be a sensor, the second one can be another sensor (passive vision) or a light source (active vision), and the third point is a point of the 3D object. Triangulation methods include structured light, shape from stereo , and shape from motion.

2.1 Shape from Stereo

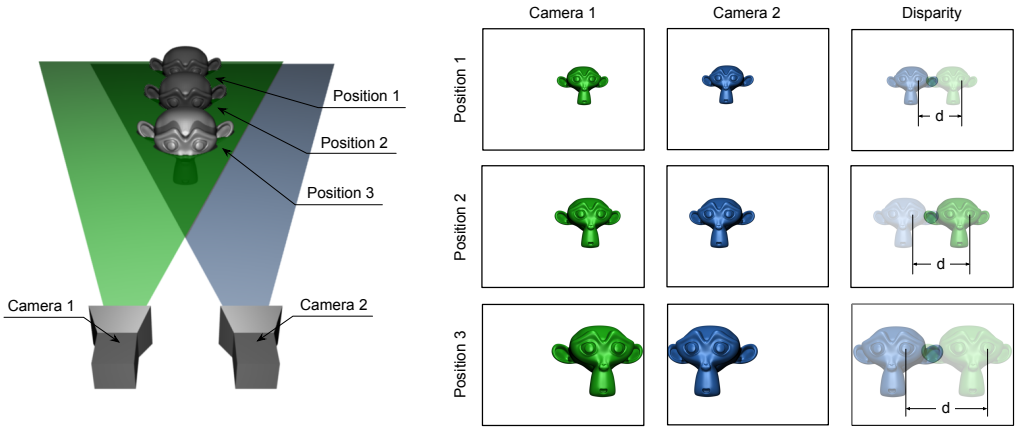


Figure 2.2: Illustration of the main idea of stereo vision.

One of the earliest approaches to capture depth information is the so called shape from stereo approach, which is a triangulation technique. It refers to the ability to infer information on the 3D structure of a scene from two or more images taken from different viewpoints [15]. Furthermore, it relies on the difference or disparity between the recorded position of each target point on the images (cf Fig. 2.2). The magnitude of this disparity is directly related to the feature's distance from the imaging sensor. The two main problems related to stereo vision are the correspondence problem and the reconstruction problem.
The correspondence problem is the problem of determining which pixel in the first image corresponds to which pixel in the second image.
The reconstruction problem deals with the calculation of 3D point locations and 3D structures of an observed scene, given a number of corresponding points as well as a camera model.

2.2 Shape from Motion

The basic concept of shape from motion [30, 25] is depth estimation of object points using the motion of either the camera or the object itself. The main difference to shape from stereo is the use of images, taken not only from different viewpoints, but also from different points in time. Shape from motion normally uses a monocular sequence of closely sampled images taken over a period of time, where either the object or the camera has been moved. Therefore the challenging part in this method is given by the calculation of the camera trajectory.

2.3 Photometric Stereo

Figure 2.3: Illustration of photometric stereo, which is a technique for estimating surface orientations by observing a scene from the same viewpoint at different illumination conditions.

Photometric stereo [31] is another technique for computing depth information from images. It is based on the Lambertain surface model, which assumes that the observed intensity of the surface does not change according to the view point. This technique estimates the surface normals of an object by observing the surface under illumination from different directions, but from the same viewpoint. The special case where there is only one image available is known as shape from shading [16]. The overall accuracy of the photometric stereo approach can be improved by a more detailed analysis of lighting conditions, such as shadows, inter-reflections and highlights. Photometric stereo only needs a single camera and a few light sources. Therefore, the main advantage is the inexpensive setup. However, photometric stereo is highly sensitive to ambient light. Hence usually dark room conditions are required for accurate results.

2.4 Shape from (De)Focus

Shape from (de)focus [24, 7, 3, 23] is another approach of estimating the 3D surface of a scene from a set of two or more images. The images are obtained from the same point of view, but with different camera parameters. Typically, the focal settings or the axial position of the image plane are changed.
In the depth from focus case, the in-focus-plane is moved. At each position the depth for focused target points (points on the in-focus-plane) can be computed. Therefore, the problem

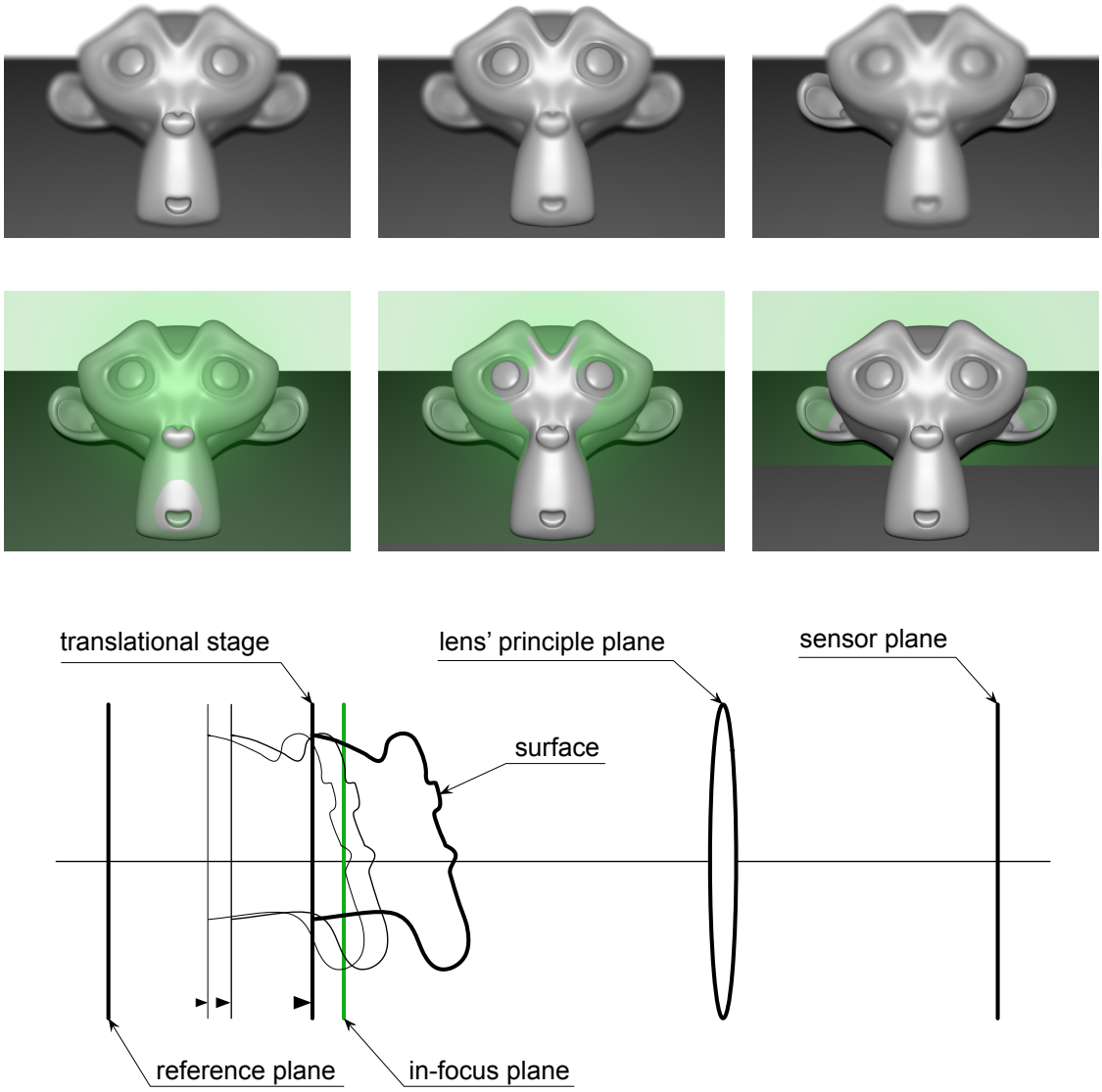


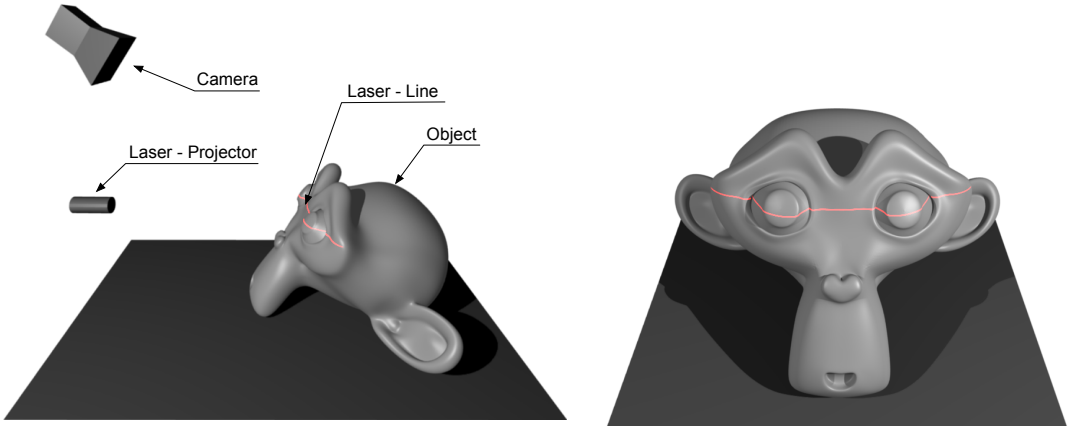
Figure 2.4: Illustration of depth from focus. The three images in the first row are acquired using different focal settings. The images in the second row show the according in-focus planes in green. The sketch below shows the whole setup from the side view.

of measuring focus needs to be solved (cf Fig. 2.4).
Contrary, in the depth from defocus case the depth information is obtained by measuring the diameter of the target feature's defocus blur spots. Depth from defocus (DFD) consists of reconstructing the distances to targets in an observed scene by modeling the effect that the camera's focal parameters have on two or more images. DFD techniques are usually passive and require only a single camera. The most difficult part of the method is the deconvolution of the defocus operator from the observed scene. Compared to common triangulation methods,

DFD methods are generally more reliable and robust, because they rely on more points in depth estimation.

2.5 Shape from Structured Light

The principle of structured light is, that by projecting a narrow band of light onto a 3D surface, a line of illumination is produced that appears distorted from other perspectives than that of the projector. Therefore, this line can be used for an exact geometric reconstruction of the 3D surface. Based on this principle, structured light scanners project 1D or 2D light patterns onto the target and extract pattern deformations.



(a) Setup of a laser scanner.

(b) Camera field of view.

Figure 2.5: Illustration of a laser scanner. Figure (a) shows the whole setup consisting of an object, a line-laser and a camera. Figure (b) shows the camera field of view.

An example for a 1D pattern is given by a projected laser line, which leads to a laser scanning system. Such an system is amongst the most accurate 3D imaging systems available, and it is also the most widely used triangulation-based 3D scanner, because of its optical and mechanical simplicity and cost. In this system, a laser sheet is swept across the target. The projection on the target surface is recorded by the camera (cf Fig. 2.5). Depending on how far away the laser strikes the target, the laser dots will appear on different locations in the camera's field of view. Similar to the depth from stereo approach, the relative position and orientation of the camera and the laser projector are known. Therefore, the distances between the sensor and the target features along the laser line can be determined uniquely.
The main disadvantages of these slit scanners include finding a compromise between the field of view and depth resolution, and their relatively poor immunity to ambient light.
An example of a 2D pattern is a line stripe pattern. Although, many other variants of patterns are possible, patterns of parallel stripes are widely used. The most popular methods for 2D pattern projection use binary coded patterns, which lead to gray-coded binary images, as

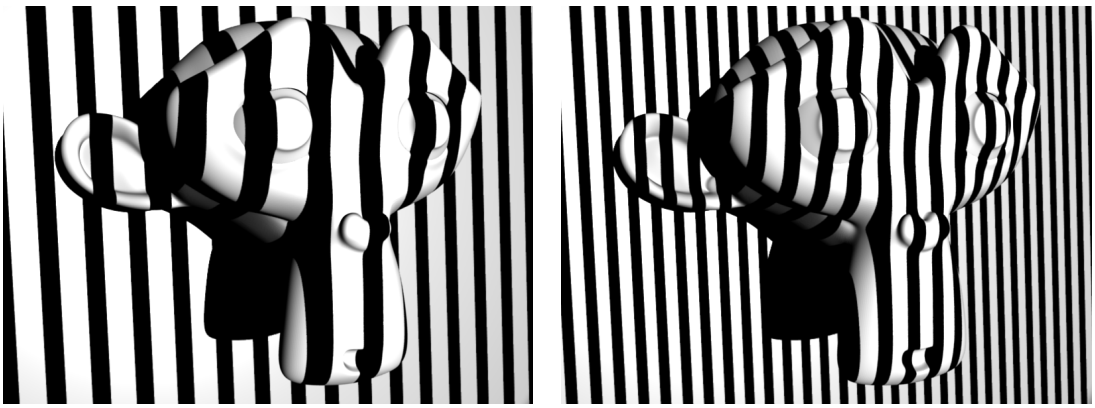


Figure 2.6: Illustration of gray-coded structured light. Here, two different linear patterns are projected on an object. The way that these patterns deform, when striking the object, allows to calculate depth information.

shown in Figure 2.6. These gray-coded binary images use multiple frames with increasing resolution to encode a pixel with the corresponding depth on the image.
Reasons for the popularity of coded patterns include the availability of low-cost projectors, the relatively fast acquirement of 3D structures, and the possibility of acquiring 3D structures without complex mechanical scanning devices. Furthermore, the speckle noise associated with lasers is reduces due to the use of incoherent light, which consequently provides better surface smoothness. However, compared to laser stripe scanners the depth of view is smaller as well as the absolute accuracy of the 3D volume.

2.6 Pulsed Time-of-flight

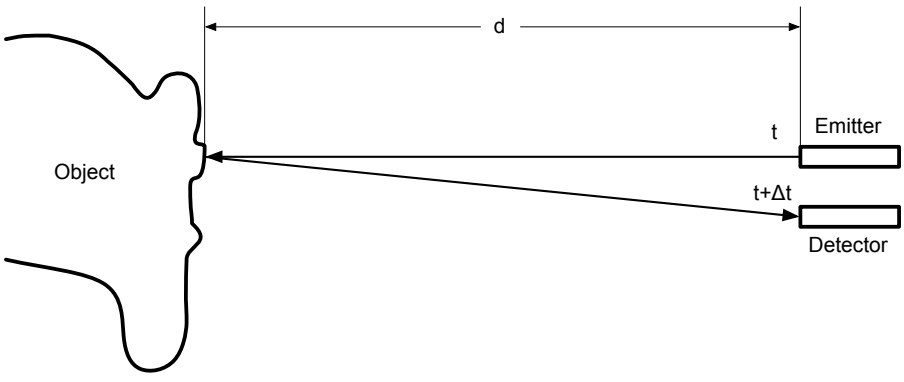


Figure 2.7: Pulsed time-of-flight rangefinder principle. Here the emitter produces a laser light or ultrasonics at time t. This energy emission is than reflected by the object. Depending on distance d, there will be a delay Δt, before the reflected energy emission is recognized by the detector. Depth is measured, based on the delay.

The two main representatives of flight range scanners [6] are ultrasonic rangefinders and laser rangefinders, which use laser light and ultrasonics, respectively, to probe an object. Therefore, the most relevant parameters are the speed of light and the speed of sound.
Laser rangefinders are able to acquire large objects with high accuracy. They calculate the distance of a surface point by measuring the round-trip time of a pulse of light. More precise, the laser is used to emit a pulse of light. The amount of time, elapsed before the reflected light is recognized by the detector, is measured. Since the speed of light c is known, the round-trip time Δt can be used to determine the target distance $d = (c \cdot \Delta t)/2$. Thus, the accuracy depends on the precision of the round-trip time measurement. The main advantage of time-of-flight systems is the constant scanning accuracy over a wide range, regardless of the distance to the object. Compared to triangulation-based systems, time-of-flight systems offer a greater operating range, which is especially useful for outdoor navigation tasks. However, the laser rangefinder only detects the distance of one point at a time. Therefore, the main disadvantage of such systems is the prolonged operation time, required to scan an entire object. Moreover, the scanners are usually larger than triangulation-based scanners, and they only capture the geometry and not the object's texture.

2.7 Wavefront Sampling

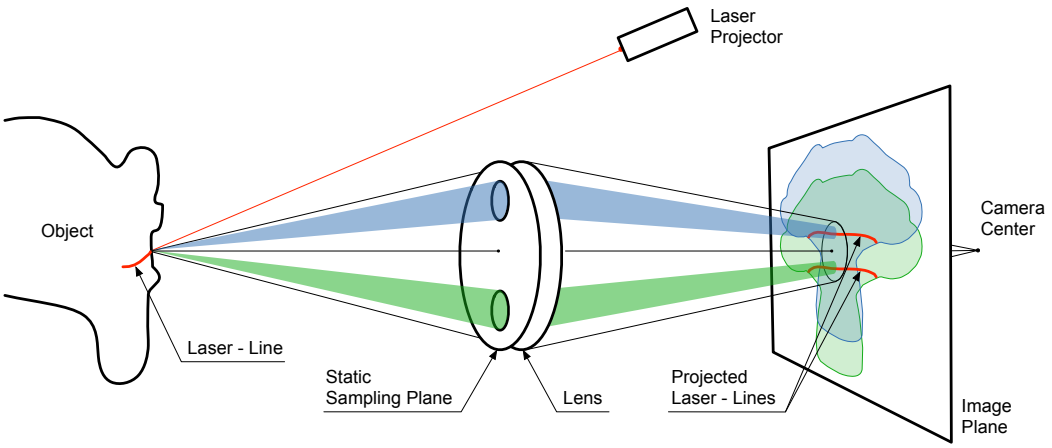


Figure 2.8: The Figure illustrates the main idea of the BIRIS range sensor. This sensor is based on the static wavefront sampling (SWS) approach. Two diametrically opposed apertures are used to sample the wavefront. This sampling pattern results in the projection of two quasi-in-focus images of the target on the image plane, where the distance between the images is used to calculate the depth information.

In a wavefront sampling system only special parts of the wavefront are allowed to reach the image plane. There are generally two types of wavefront sampling system, the static wavefront sampling (SWS) and the active wavefront sampling (AWS) approach.
In the SWS case (cf Fig. 2.8), a sampling mask with at least two off-axis apertures is used to sample the wavefront. This mask produces multiple images of an object-point on the image

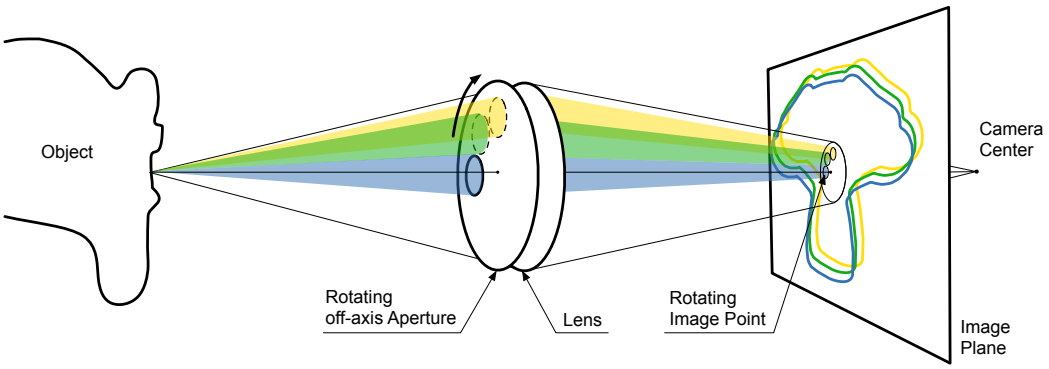


Figure 2.9: Illustration of the active wavefront sampling approach (AWS). Here a single off-axis aperture is rotated around the optical axis. Thus, a target point's image will appear to rotate on the image plane. The fact, that at each aperture position only a single image is recorded on the image plane eliminates any kind of overlapping problem.

plane. The distance between these imaged object points is then used to calculate the depth information. One implementation of a two aperture SWS approach is the so called BIRIS range sensor, developed at National Research Council, Canada [26, 4]. The main components of this sensor are: a sampling mask with two apertures, a camera lens, and a standard CCD camera. The two aperture mask replaces the iris of a standard camera lens and therefore the sensor was named BI-IRIS.
Depth calculation in the BIRIS sensor is accomplished as follows: A laser line produces two images on the sensor plane. By measuring the distance between the two lines the depth of the target points illuminated by the laser is calculated. Similar to a regular laser scanner, the target needs to be gradually scanned in order to obtain the entire depth map.
It is possible to use more than two sampling positions for the depth estimation. Such a sampling mask with more than two apertures would result in images with a corresponding number of lines. By measuring the displacements between the various lines depth can be calculated with a higher potential depth estimation accuracy. Thus, on the one hand, more sampling positions would increase the accuracy, but on the other hand it would also cause confusion due to possibly overlapping lines.
The main disadvantage of SWS is the fact, that the images are recorded on the same image plane, which can cause overlapping problems. The active wavefront sampling (AWS) approach (cf Fig. 2.9) is a further development of the SWS approach and provides a solution to the overlapping problem. In an AWS system a single off-axis aperture is moved from one position to the next. At each position a single quasi-in-focus image is recorded without multiple image overlap. Moreover, images are recorded from multiple perspectives, which increases the accuracy of the calculated depth maps. Due to the relatively simple mechanical implementation the off-axis aperture is usually rotated around the optical axis, but this is not the only possible path that can be used. In theory depth can be recovered as long as the aperture path is known.

Chapter 3

Background

Contents

This chapter provides relevant background information. First, in Section 3.1 a short mathematical recap regarding calculus of variations is presented. Section 3.3 presents an overview of optical flow concepts, which are used in the active wavefront sampling (AWS) approach to track the features from one sampling position to the next. Section 3.4 provides a short review about robust estimation. To clearly understand the principle of AWS one first has to understand some basic concepts about the theory of defocus. Thus, Section 3.5 provides necessary background information regarding this topic. Among other things, this section describes how an out-of-focus point is encoded by its blur spot diameter on the image plane. Section 3.6 introduces the main idea of wavefront sampling and explains how the blur-spot-diameter of a DFD system can be calculated using a static wavefront sampling (SWS) or active wavefront sampling (AWS) mask. This section also gives some basic information about size and placement of the sampling mask as well as an analysis of the depth sensitivity of an AWS based system. Further, this section also presents current 3D reconstruction algorithms using the AWS approach, presented by Frigerio [10]. First, Section 3.6.7 describes Frigerio's multi image AWS algorithm. Second, Section 3.6.8 describes an alternative version of the Frigerio multi image approach, which makes use of long spatio-temporal filter.

3.1 Calculus of Variations

Calculus of variations [33, 9, 12] is a field of mathematics, that deals with the problem of minimizing expressions of the form

$$F\left[y(x)\right] = \int_a^b f(x, y(x), y'(x))\,\mathrm{d}x, \tag{3.1}$$

where $f : [a,b] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is given and one seeks a function $y : [a,b] \to \mathbb{R}^d$. The fundamental result of the calculus of variations is the following theorem.

Theorem 1 *(Euler-Lagrange Equation)*
If $y(x)$ is a curve in $C^2_{[a,b]}$ that minimizes the functional

$$F\left[y(x)\right] = \int_a^b f(x, y(x), y'(x))\,dx, \tag{3.2}$$

then the following equation must be satisfied

$$\frac{\partial f}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}x}\left(\frac{\partial f}{\partial y'}\right) = 0. \tag{3.3}$$

This equation is referred to as Euler-Lagrange Equation.

To proof Teorem 1 we need the Leibniz rule, and the following Lemma.

Lemma 1
Let $M(x)$ be a continuous function on the interval $[a,b]$. If for any continuous function $h(x)$ with $h(a) = h(b) = 0$

$$\int_a^b M(x)h(x)\,dx = 0, \tag{3.4}$$

then $M(x)$ is zero almost everywhere.

Proof 1
First one chooses $h(x) = -M(x)(x-a)(x-b)$. Thus, $h(x)$ is continuous because M is continuous. Moreover, $M(x)h(x) \geqslant 0$ for $x \in [a,b]$. If the definite integral of a non-negative function is zero, then the function itself is zero almost everywhere. Thus, one obtains

$$0 = M(x)h(x) = \left[M(x)\right]^2\left[-(x-a)(x-b)\right]. \tag{3.5}$$

Equation 3.5 and the fact that $[-(x-a)(x-b)] > 0$ for $x \in (a,b)$ implies that $\left[M(x)\right]^2 = 0$ on $[a,b]$. Therefore, $M(x) = 0$ on $[a,b]$.

□

Proof 2 *(Euler-Lagrange Equation)*
Suppose $y(x)$ is a curve minimizing the functional F. Thus, for any curve $g(x)$, $F\left[y(x)\right] \leqslant F\left[g(x)\right]$. Next, one constructs a variation of $y(x)$ as follows:

$$y_s(x) = y(x) + s\, h(x), \tag{3.6}$$

where $h(x) \in C^2_{[a,b]}$ *with* $h(a) = h(b) = 0$ *and* s *is close to null. Now one defines the function*

$$H(s) = F\left[y_s(x)\right]. \tag{3.7}$$

Due to the fact that $y(x)$ *minimizes* $F\left[y(x)\right]$ *it follows that* 0 *minimizes* $H(s)$*. From ordinary calculus one obtains* $H'(0) = 0$ *because* $H(0)$ *is a minimum value for* H*. Using the Leibniz rule one obtains*

$$\frac{\mathrm{d}}{\mathrm{d}s} H(s) = \frac{\mathrm{d}}{\mathrm{d}s} \int_a^b f(x, y_s(x), y_s'(x))\, \mathrm{d}x = \int_a^b \frac{\partial}{\partial s} f(x, y_s(x), y_s'(x))\, \mathrm{d}x. \tag{3.8}$$

Next, the chain rule is used within the integral.

$$\frac{\partial}{\partial s} f(x, y_s(x), y_s'(x)) = \frac{\partial f}{\partial x}\frac{\partial x}{\partial s} + \frac{\partial f}{\partial y_s}\frac{\partial y_s}{\partial s} + \frac{\partial f}{\partial y_s'}\frac{\partial y_s'}{\partial s} = \frac{\partial f}{\partial y_s} h(x) + \frac{\partial f}{\partial y_s'} h'(x) \tag{3.9}$$

Incorporating this into Equation 3.8 leads to

$$H'(s) = \int_a^b \left(\frac{\partial f}{\partial y_s} h(x) + \frac{\partial f}{\partial y_s'} h'(x) \right) \mathrm{d}x \tag{3.10}$$

By setting $s = 0$ *one obtains*

$$0 = \int_a^b \left(\frac{\partial f}{\partial y} h(x) + \frac{\partial f}{\partial y'} h'(x) \right) \mathrm{d}x. \tag{3.11}$$

By applying integration by parts to the second term in Equation 3.11 one finally receives

$$0 = \int_a^b \left[\frac{\partial f}{\partial y} h(x) - \frac{\mathrm{d}}{\mathrm{d}x} \left(\frac{\partial f}{\partial y'} \right) \right] h(x)\, \mathrm{d}x. \tag{3.12}$$

This procedure works for any function $h(x)$ *with* $h(a) = h(b) = 0$*. Therefore, one can use Lemma 1 to obtain the Euler-Lagrange Equation*

$$0 = \frac{\partial f}{\partial y} - \frac{\mathrm{d}}{\mathrm{d}x} \left(\frac{\partial f}{\partial y'} \right). \tag{3.13}$$

□

Example

Finding the real-valued function $y(x)$ on the interval $[a, b] \subset \mathbb{R}$, such that the arc-length between the points $(a, y(a))$ and $(b, y(b))$ is minimized, is one of the standard examples to illustrate such a problem (cf Fig. 3.1). Here the length of the graph $y(x)$ is given by

$$L(y) = \int_a^b \sqrt{1 + y'(x)^2}\, \mathrm{d}x. \tag{3.14}$$

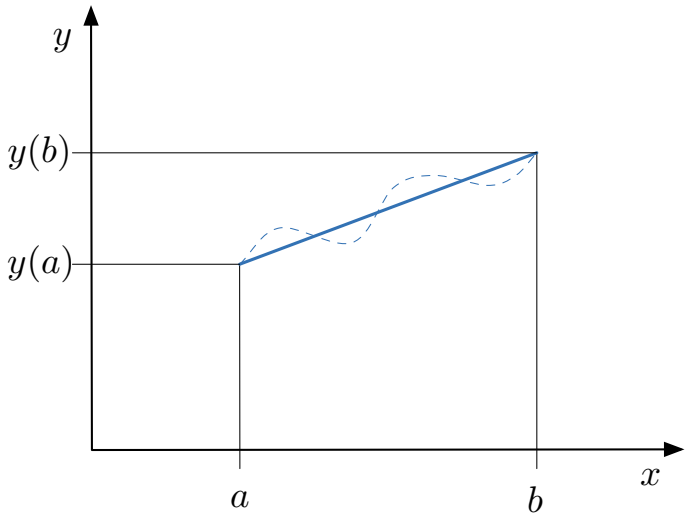


Figure 3.1: Illustration of the functional $L(y)$. Here the dashed blue graph depicts an arbitrary function on the interval $[a, b] \subset \mathbb{R}$ connecting the points $(a, y(a))$ and $(b, y(b))$. The solid blue graph depicts the stationary function of $L(y)$, which is of course a straight line.

The partial derivatives of L are as follows:

$$\frac{\partial f(x, y, y')}{\partial y'} = \frac{y'}{\sqrt{1 + y'^2}} \quad \text{and} \quad \frac{\partial f(x, y, y')}{\partial y} = 0. \tag{3.15}$$

By substituting the results from 3.15 into the Euler-Lagrange equation (3.3), one obtains

$$\frac{\mathrm{d}}{\mathrm{d}x} \frac{y'(x)}{\sqrt{1 + y'(x)^2}} = 0, \tag{3.16}$$

which indicates that

$$\frac{y'(x)}{\sqrt{1 + y'(x)^2}} = C, \tag{3.17}$$

where C is a constant. As a consequence, $y'(x)$ is constant too. This implies, that the according graph is a straight line.

3.2 Visual Motion

The world changes with time, so that in any realistic vision problem we need to understand and deal with motion. Images are obtained by projecting the 3D world onto a 2D light sensing surface, for example a piece of film, an array of light sensors or photoreceptors in the human eye. By moving an object in the 3D world, the 2D projection of that object will move also. This movement of the 2D projection is referred to as image velocity or visual motion. The visual motion field provides a valuable source of information for analyzing the observed scene in terms of objects, their motion in space, and also their 3D structure.

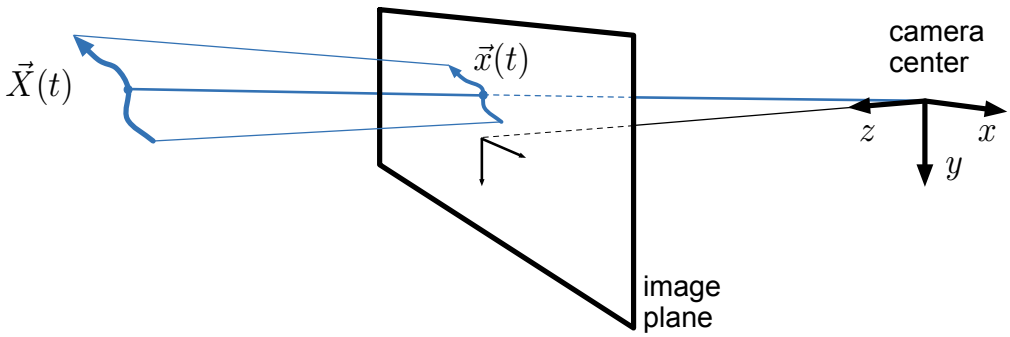


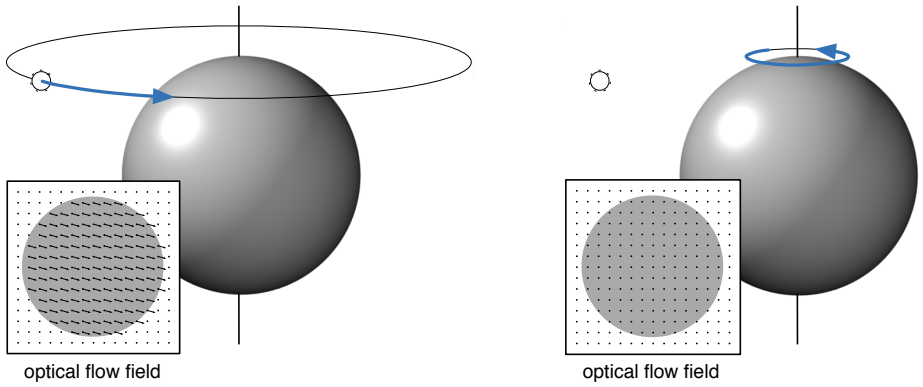
Figure 3.2: Illustration of image motion. A 3D surface point follows a space-time path $\vec{X}(t)$. Projecting this path onto the image plane produces a 2D path $\vec{x}(t)$, where the instantaneous 2D velocity is called image motion.

3.3 Optical Flow

(a) First frame. (b) Color-coded optical flow. (c) Second frame.

Figure 3.3: Illustration of the optical flow principle. Figure (a) and (c) show two consecutive frames. Figure (b) shows the color-coded optical flow field, calculated using a global optical flow method [32]. Direction and magnitude of the flow vectors are visualized by the color and the intensity, respectively.

A fundamental problem of image processing is the measurement of motion or image velocity. Motion is an important part of our visual experience and supports a wide range of visual tasks, such as 3D shape acquisition, object recognition and scene understanding.
The goal of optical flow estimation is to approximate the 2D velocity field or also called 2D motion field. The motion field is a projection of the 3D velocities of surface points onto the image plane. More precisely, each point on a 3D surface moves along a path $\vec{X}(t)$. By projecting this path onto the image plane it produces a 2D path $\vec{x}(t)$, where the instantaneous direction $d\vec{x}(t)/dt$ is called velocity.
Note, optical flow is a radiometric concept. It explains brightness variation, while motion



(a) Optical flow without motion.

(b) Motion without optical flow.

Figure 3.4: Illustration of the difference between visual motion and optical flow. In Figure (a) a static sphere is illuminated by a moving light source, which is an example for optical flow without motion. In Figure (b) a spinning sphere is illuminated by a static light source, which is an example for motion without optical flow.

is geometric. Thus, optical flow and motion are two different things. There may be motion without optical flow (consider e.g. a spinning sphere of uniform appearance) and optical flow without motion (consider e.g. a scene illuminated by a moving light source). In the taxonomy of computer vision problems, the optical flow problem belongs to the class of correspondence-problems. Given two or more images of the same scene, the correspondence-problem is to find an optimal relation between corresponding image data.

3.3.1 Basic Observations

The calculation of optical flow is a notoriously error-prone problem. There are several situations that occur in real-world scenes causing trouble for most algorithms. Some of those problems are mentioned below.
In some situations there is insufficient information to calculate the optical flow (cf Fig. 3.5). By considering, for example an untextured flat surface, we cannot determine motion at all, because the image remains constant over time. Due to the fact that we study techniques that estimate the motion by considering small regions, this problem will occur whenever the considered region has a uniform image brightness.
Another problem occurs by looking at a striped pattern. In this case one cannot determine the velocity along the direction of the stripes, because, if the pattern slides along the direction of the stripes, the image will not change. In literature this problem is usually referred to as the aperture problem.
Most algorithms for optical flow estimation are based on the assumption that changes in the image intensity are due only to motion. Unfortunately, in real-world images this assumption is frequently violated. For example, if lighting changes occur, all intensities in the image

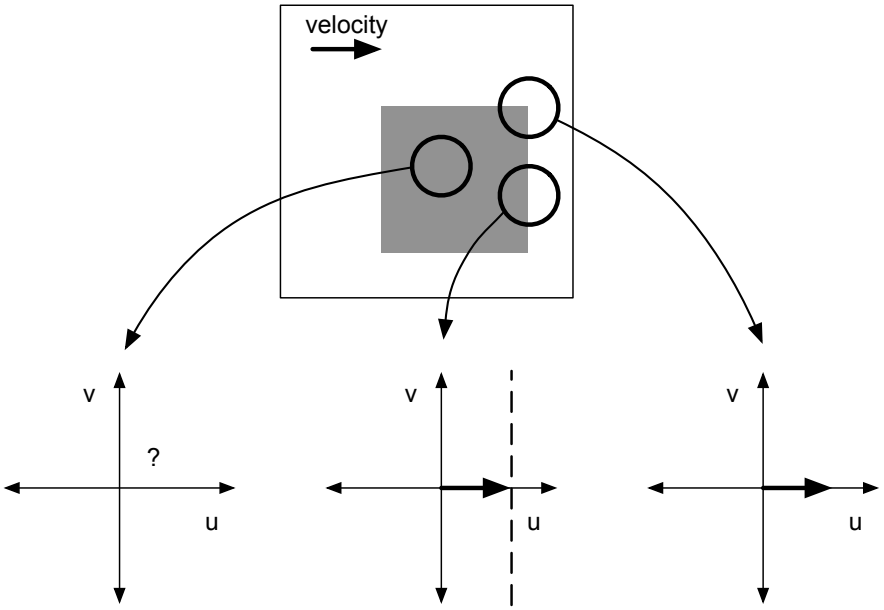


Figure 3.5: Illustration of motion situations for the two types of singularities and the non-singularity situation. In the center of the square the motion is completely unconstrained. At the edge the intensity changes are consistent forming a 1D set of velocities (dashed line). Here we cannot determine the motion along the edge, only the motion perpendicular to the edge can be determined. At the corner a unique velocity can be calculated. [29]

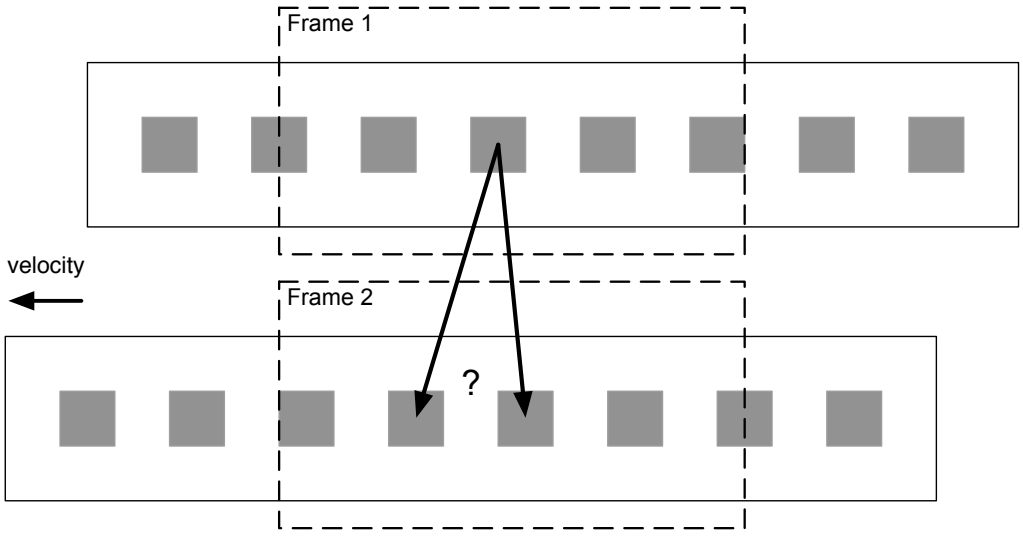


Figure 3.6: Illustration of temporal aliasing. Two frames are shown, where a series of squares is moving (to the left). If the motion is slow (smaller than half the distance between two consecutive squares), the motion will be recognized correctly. If the motion is faster than half the distance between two consecutive squares, the opposite motion will be estimated.

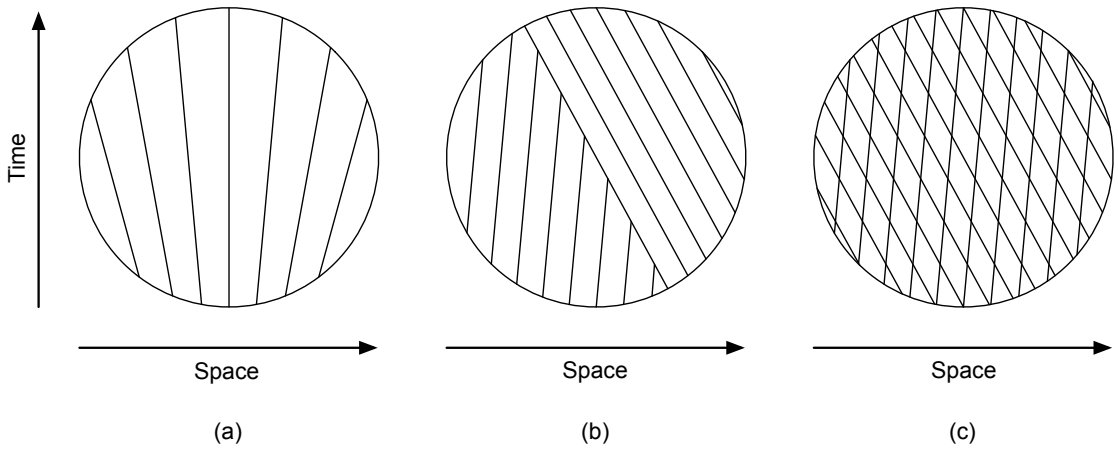


Figure 3.7: Space-time illustration of three types of non-translational motion (dilation, occlusion, and transparency). Figure (a) shows the situation of a dilation. Figure (b) illustrates the case of an occlusion boundary, where two patterns move towards each other, and the right occluding the left. Finally, Figure (c) shows rightward and leftward moving transparent patterns. [29]

will change either. Or by considering orientation changes of surfaces relative to the camera, the reflected light towards the camera will also change. In such situations an optical flow algorithm based on the brightness conservation assumption will fail.
Owing to the fact that the input to an optical flow estimation system is sampled with respect to time, it is usually difficult to estimate large motions. This problem is clarified by the following example: Consider a series of squares moving to the left (cf Fig. 3.6). If the velocity is low then the motion will be correctly recognized by a local motion algorithm. However, if the velocity is high, there will not be a correct estimation of the motion any longer. For some velocities, the squares will even seem to move backwards (to the right). This problem is called temporal aliasing.
Almost all optical flow approaches are based on the assumption that motion is translational and hence a single velocity vector is sufficient to describe motion. But this assumption is frequently violated in real-world scenes as well. Typical examples are due to rotation, dilation or motion of non-rigid objects (e.g. fluids or elastic materials). This will lead to a small motion estimation error as long as the regions over which motion is being estimated are small compared to the deviations from translational motion.
More severe problems occur at occlusion boundaries and in the presence of transparency or highlights. In these situations motion cannot be properly described by a single displacement vector.

3.3.2 Differential Techniques

Due to the fact that velocity is a differential quantity, it is intuitive to consider derivative measurements for estimation. Differential techniques, also known as gradient techniques, use

spatiotemporal derivatives of image intensity to compute the image velocity. This is, in many ways, the simplest and also the most elegant approach to estimate the motion field.

3.3.2.1 Basic Gradient-Based Estimation

Gradient-based methods assume that pixel intensities are translated from one image to the next, with respect to the brightness constancy assumption

$$I(\vec{x}, t) = I(\vec{x} + \vec{u}, t + \delta t), \tag{3.18}$$

where $I(\vec{x}, t)$ is the image intensity as a function of space $\vec{x} = (x, y)^T$ and time t, $\vec{u} = (u, v)^T$ is the 2D velocity and δt is a small temporal increment. Without loss of generality, we will assume $\delta t = 1$. Note, that the brightness constancy does not hold exactly. It is linked to the assumption that surface radiance remains constant with respect to time, as mentioned in the previous section.

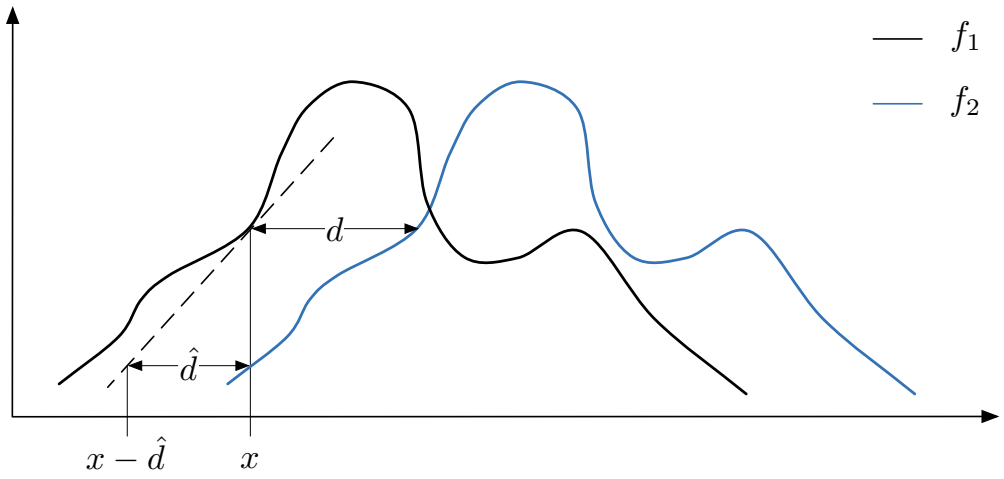


Figure 3.8: This illustration shows how the gradient constraint relates the displacement of a signal to its temporal difference and spatial derivatives (slope). For a nonlinear signal, the difference divided by the slope gives an approximation $\hat{d}$ to the true displacement d.

First of all, the 1D case will be considered. There are two 1D signals f_1 and f_2, where the second signal is a translated version of the first one, i.e. $f_2(x) = f_1(x - d)$ where d denotes the translation (cf Fig. 3.8). Then f_1 is linearized by using a Taylor series expansion of $f_1(x - d)$ about x:

$$f_1(x - d) = f_1(x) - df_1'(x) + O(d^2 f_1''). \tag{3.19}$$

Using this Taylor series expansion the difference between the signals can be written as follows:

$$f_1(x) - f_2(x) = df_1'(x) + O(d^2 f_1''). \tag{3.20}$$

By ignoring second- and higher-order terms, one receives the following approximation to d:

$$\hat{d} = \frac{f_1(x) - f_2(x)}{f_1'(x)}. \tag{3.21}$$

The 1D case can be generalized to 2D in a straightforward way. Again, it is assumed that the displaced image is approximated by a first-order Taylor series expansion.

$$I(\vec{x} + \vec{u}, t + 1) \approx I(\vec{x}, t) + \vec{u} \cdot \nabla I(\vec{x}, t) + I_t(\vec{x}, t), \tag{3.22}$$

where $\nabla I = (I_x, I_y)$ denote spatial and I_t temporal partial derivatives of the image I, and $\vec{u} = (u, v)^T$ denotes the 2D velocity. By substituting this linear approximation into the brightness constancy assumption (Equation 3.18) one obtains the so called gradient constraint equation:

$$\nabla I(\vec{x}, t) \cdot \vec{u} + I_t(\vec{x}, t) = 0. \tag{3.23}$$

The gradient constraint equation can also be derived more generally from an estimation of 2D paths $\vec{x}(t)$ along which intensity is conserved:

$$I(\vec{x}(t), t) = c, \tag{3.24}$$

where c is a constant. By using the temporal derivative and the chain rule for differentiation, one again obtains the gradient constraint equation:

$$\begin{aligned} &\frac{d}{dt} I(\vec{x}(t), t) = \frac{d}{dt} c \\ &\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t}\frac{dt}{dt} = 0 \\ &\nabla I \cdot \vec{u} + I_t = 0, \end{aligned}$$

with $\vec{u} = (dx/dt, dy/dt)^T$. The gradient constraint equation (Equation 3.23) defines the relationship between the image intensities and the optical flow, and is fundamental to all differential techniques.

It is one equation with two unknowns and therefore it constrains the optical flow $\vec{u}$ to a one parameter family of velocities along a line in velocity space (cf Fig. 3.9) [19]. The velocity (u, v) has to lie along a line perpendicular to ∇I and its perpendicular distance from the origin is $|I_t|/\|\nabla I\|$. Therefore, the locus of solutions to the gradient constraint equation constitutes a line in the 2D velocity space:

$$\vec{u} = -\frac{I_t \cdot \nabla I}{\|\nabla I\|^2} + \alpha \frac{1}{\|\nabla I\|} \begin{pmatrix} -I_y \\ I_x \end{pmatrix}, \tag{3.25}$$

where α is a variable, parameterizing the line. It should be mentioned, that the first term in Equation 3.25 is called normal velocity.

Due to the fact, that the gradient constraint equation is one equation with two unknowns,

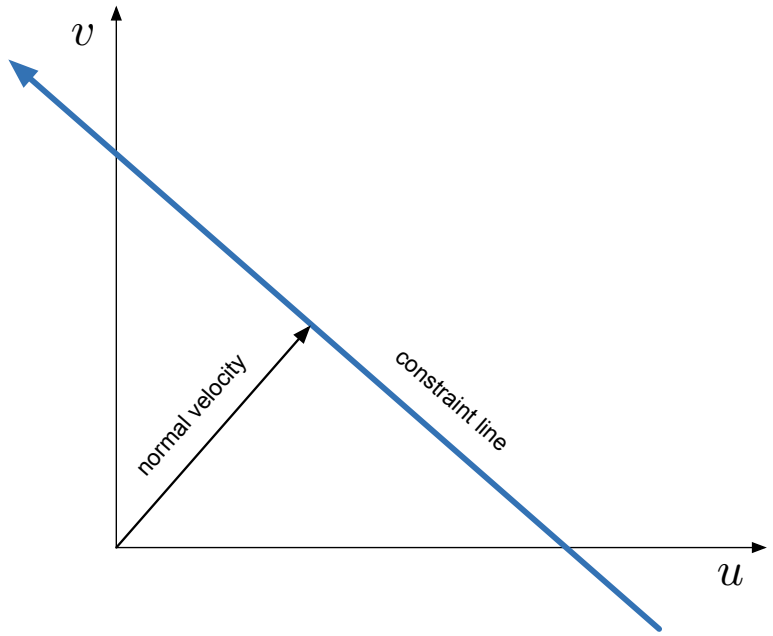


Figure 3.9: The set of velocities that satisfy the gradient constraint equation (3.23) constitutes a line in the 2D velocity space. Furthermore, this line has to be perpendicular to the brightness gradient vector. The distance of this line from the origin equals $|I_t|/\|\nabla I\|$. Thus, the so called normal velocity is $-I_t \nabla I/\|\nabla I\|^2$. [19]

relying the calculation only on this constraint leads to an under-constrained or ill-posed[1] problem. In order to obtain a well-posed problem, additional modeling assumptions are needed.

3.3.2.2 Local Methods

Due to the aperture problem, it is not possible to calculate the flow vector from a single gradient constraint equation. Therefore, further constraints are needed. One idea is using multiple illuminations, where more equations under different illuminations are obtained.
Another idea is to use an intensity derivative conservation assumption, where the gradient constraint equation is derived with respect to x, y and/or t. This leads to the optical flow calculations based on second-order derivatives. The constraint equation including all three partial derivatives yields to:

$$\begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \\ I_{xt} & I_{xt} \end{pmatrix} \vec{u} + \begin{pmatrix} I_{xt} \\ I_{yt} \\ I_{tt} \end{pmatrix} = 0. \tag{3.26}$$

[1] Hadamard defines problems to be well-posed if a solution exists, the solution is unique, and if the solution depends continuously on the data, in some reasonable topology. If any of those conditions is violated, the problem is called ill-posed.

Equation 3.26 has three constraints for the two unknowns. Therefore it is no longer under-constrained like Equation 3.23.
Note that this method can be extended to any order of derivative. It is also reasonable to combine constraints from different orders of derivatives. E.g., one can combine the second-order constraints with the first-order constraint:

$$\begin{pmatrix} I_x & I_y \\ I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \\ I_{xt} & I_{xt} \end{pmatrix} \vec{u} + \begin{pmatrix} I_t \\ I_{xt} \\ I_{yt} \\ I_{tt} \end{pmatrix} = 0. \tag{3.27}$$

An approach that combines constraints from nearby pixels to further constrain $\vec{u}$ was suggested by Lucas and Kanade [21]. They assumed that the flow is constant in a small window, which leads to the following set of equations:

$$\begin{aligned} I_x(x_1, y_1, t)u + I_y(x_1, y_1, t)v &= -I_t(x_1, y_1, t) \\ I_x(x_2, y_2, t)u + I_y(x_2, y_2, t)v &= -I_t(x_2, y_2, t) \\ &\dots \\ I_x(x_n, y_n, t)u + I_y(x_n, y_n, t)v &= -I_t(x_n, y_n, t), \end{aligned}$$

where (x_i, y_i, t) for $1 \leq i \leq n$ are the space-time coordinates of points in the window. This leads to an overdetermined system of equations:

$$\begin{pmatrix} I_x(x_1, y_1, t) & I_y(x_1, y_1, t) \\ .. & .. \\ I_x(x_n, y_n, t) & I_y(x_n, y_n, t) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -I_t(x_1, y_1, t) \\ .. \\ -I_t(x_n, y_n, t) \end{pmatrix}, \tag{3.28}$$

which can be solved using least-squares estimation. Furthermore, they used a weighting function that determines the support of the estimator and gives more influence to constraints at the center of the neighborhood. Hence, they used a weighted least-squares fit of local first-order constraints:

$$E(u, v) = \sum_{\vec{x} \in \Omega} g(\vec{x}) \left[\nabla I(\vec{x}, t) \cdot \vec{u} + I_t(\vec{x}, t)\right]^2, \tag{3.29}$$

where $g(\vec{x})$ is the weighting function, and Ω denotes a local neighborhood of $\vec{x}$. Minimization of Equation 3.29 is done by evaluating the first derivatives with respect to the vector $\vec{u}$:

$$\begin{aligned} \frac{\partial E(u, v)}{\partial u} &= \sum_{x \in \Omega} g(\vec{x})(uI_x^2 + vI_xI_y + I_xI_t) = 0, \\ \frac{\partial E(u, v)}{\partial v} &= \sum_{x \in \Omega} g(\vec{x})(vI_y^2 + uI_xI_y + I_yI_t) = 0. \end{aligned}$$

This system of equations can be rewritten in matrix form as

$$M\vec{u} = \vec{b}, \tag{3.30}$$

where M and $\vec{b}$ are:

$$M = \begin{pmatrix} \sum g(\vec{x}) I_x^2 & \sum g(\vec{x}) I_x I_y \\ \sum g(\vec{x}) I_y I_x & \sum g(\vec{x}) I_y^2 \end{pmatrix},$$

$$\vec{b} = - \begin{pmatrix} \sum g(\vec{x}) I_x I_t \\ \sum g(\vec{x}) I_y I_t \end{pmatrix}.$$

The least squares estimate to Equation 3.30 is $\hat{\vec{u}} = M^{-1}\vec{b}$, if $rank(M) = 2$. If M is rank deficient, the problem is not properly constrained, which means that the local neighborhood Ω has insufficient information to solve the aperture problem. By increasing the size of the local interrogation area more constraints can be added, which increases the chance to find a unique solution. Note, by increasing the neighborhood, the assumption of a single flow vector for the whole neighborhood becomes increasingly questionable. One solution to this dilemma is to calculate the flow only at points where M has full rank, which results in a sparse flow field.

3.3.2.3 Global Methods

Local methods will yield poorly conditioned systems of equations in image regions of nearly uniform intensity. Global methods try to avoid this singularity problem by incorporating additional constraints or regularization terms to the energy function. As indicated by the name, global methods integrate the information from the entire image domain to infer the flow. Contrary to sparse local methods, where a dense flow field is obtained by an interpolation step, global methods directly yield a dense flow field.
The motion models are generally classified as fully parametric, quasi-parametric, and non-parametric by Bergen et al. [5].
A fully parametric flow estimation model describes the motion of individual pixels within a region via a parametric form. Therefore, parametric models are limited to applications, where the characteristics of the flow fields are known. Examples for fully parametric models include affine and quadratic flow fields.
Quasi-parametric models combine a parametric component that is valid for the entire region with a local component, which varies from pixel to pixel, like e.g. the rigid motion model.
Non-parametric models do not express the additional assumption explicitly, but use some type of regularisation term, e.g., Horn and Schunck [19] used a constraint of global smoothness of the velocity field. Hildreth [18] used a constraint of smoothness along contours. Nagel [22] suggested a global oriented-smoothness constraint, in which less smoothing is done in the direction of the gradient. These methods will be briefly discussed below.
The main disadvantage of global methods is the computational cost, which is far higher than with local methods.

Horn and Schunck [19]
Horn and Schunck presented a method for finding the optical flow, which assumes that the apparent velocity of the brightness pattern varies smoothly almost everywhere in the image. They combined the gradient constraint (Equation 3.23) with a global smoothness term,

$$\int_{\Omega} (\nabla I \cdot \vec{u} + I_t)^2 + \alpha^2 \underbrace{(\|\nabla u\|_2^2 + \|\nabla v\|_2^2)}_{\text{smoothness term}} d\Omega, \tag{3.31}$$

where α reflects the influence of the smoothness term and Ω is the image space. The minimization is done by using the calculus of variation. They obtained the following system of equations:

$$I_x^2 u + I_x I_y v = \alpha^2 \nabla^2 u - I_x I_t,$$
$$I_x I_y u + I_y^2 v = \alpha^2 \nabla^2 v - I_y I_t.$$

By using an approximation to the Laplacian they obtained

$$(\alpha^2 + I_x^2)u + I_x I_y v = (\alpha^2 \bar{u} - I_x I_t),$$
$$I_x I_y u + (\alpha^2 + I_y^2)v = (\alpha^2 \bar{v} - I_y I_t),$$

where $\bar{u}$ and $\bar{v}$ denote neighborhood averages. This system is solved for u and v , which yields to

$$(\alpha^2 + I_x^2 + I_y^2)u = +(\alpha^2 + I_y^2)\bar{u} - I_x I_y \bar{v} - I_x I_t,$$
$$(\alpha^2 + I_x^2 + I_y^2)v = -I_x I_y \bar{u} + (\alpha^2 + I_x^2)\bar{v} - I_y I_t.$$

This can finally be rewritten as:

$$(\alpha^2 + I_x^2 + I_y^2)(u - \bar{u}) = -I_x(I_x \bar{u} + I_y \bar{v} + I_t),$$
$$(\alpha^2 + I_x^2 + I_y^2)(v - \bar{v}) = -I_y(I_x \bar{u} + I_y \bar{v} + I_t).$$

Due to the fact that it would be very costly to solve this system by using standard methods, like e.g. Gauss-Jordan elimination, an iterative method is used. They compute a new set of velocities (u^{n+1}, v^{n+1}) from the derivatives and the averages of (u^n, v^n):

$$u^{n+1} = \bar{u}^n - \frac{I_x(I_x \bar{u}^n + I_y \bar{v}^n + I_t)}{\alpha^2 + I_x^2 + I_y^2},$$
$$v^{n+1} = \bar{v}^n - \frac{I_y(I_x \bar{u}^n + I_y \bar{v}^n + I_t)}{\alpha^2 + I_x^2 + I_y^2}.$$

Nagel [22]
Nagel was one of the first who used second-order derivatives to compute optical flow. Moreover, he suggested an oriented-smoothness constraint in which smoothness is not imposed

across steep intensity gradients (edges). The problem is formulated by the following functional:

$$\int_{\Omega} (\nabla I \cdot \vec{u} + I_t)^2 + \frac{\alpha^2}{\|\nabla I\|_2^2 + 2\delta}((u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2 + \delta(u_x^2 + u_y^2 + v_x^2 + v_y^2))d\Omega \quad (3.32)$$

Using Gauss-Seidel iterations, the solution can be computed as follows:

$$u^{k+1} = \xi(u^k) - \frac{I_x(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{I_x^2 + I_y^2 + \alpha^2},$$
$$v^{k+1} = \xi(v^k) - \frac{I_y(I_x\xi(u^k) + I_y\xi(v^k) + I_t)}{I_x^2 + I_y^2 + \alpha^2},$$

where k is the iteration number, and $\xi(u^k)$ and $\xi(v^k)$ are given by

$$\xi(u^k) = \bar{u}^k - 2I_x I_y u_{xy} - q^T(\nabla u^k),$$
$$\xi(v^k) = \bar{v}^k - 2I_x I_y v_{xy} - q^T(\nabla v^k),$$

where

$$q = \frac{1}{I_x^2 + I_y^2 + 2\delta}\nabla I^T \left[\begin{pmatrix} I_{yy} & -I_{xy} \\ -I_{xy} & I_{xx} \end{pmatrix} + 2 \begin{pmatrix} I_{yy} & I_{xy} \\ I_{xy} & I_{xx} \end{pmatrix} W \right],$$

u_{xy} and v_{xy} are partial derivatives of $\vec{u}^k$, $\bar{u}^k$ and $\bar{v}^k$ are local neighborhood averages of u^k and v^k and W is the weight matrix

$$W = (I_x^2 + I_y^2 + 2\delta)^{-1} \begin{pmatrix} I_y^2 + \delta & -I_x I_y \\ -I_x I_y & I_x^2 + \delta \end{pmatrix}.$$

Hildreth [18]

Due to the aperture problem, local optical flow techniques provide, in the case of contours, only the component of motion in the direction perpendicular to the orientation of the contour. The velocity component in the direction of the contour cannot be detected. More precise, the 2D velocity field along a contour may be described by the vector function $V(s)$, where s denotes arclength. $V(s)$ can be decomposed into components tangent and perpendicular to the contour, as shown in Figure 3.10. $u^{\top}(s)$ and $u^{\perp}(s)$ are unit vectors in the directions tangent and perpendicular to the curve, and $v^{\top}(s)$ and $v^{\perp}(s)$ denote the two components [18]:

$$V(s) = v^{\top}(s)u^{\top}(s) + v^{\perp}(s)u^{\perp}(s).$$

$v^{\perp}(s)$, $u^{\top}(s)$, and $u^{\perp}(s)$ are given directly by initial measurements from the observed images.

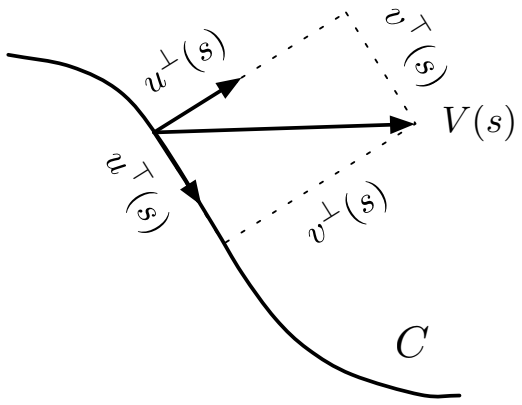


Figure 3.10: Decomposition of the velocity $V(s)$. The velocity vector $V(s)$ is decomposed into components perpendicular and tangent to the curve C. The unit direction vectors are represented by $u^{\top}(s)$ and $u^{\perp}(s)$. $v^{\perp}(s)$ and $v^{\top}(s)$ represent the two velocity components.

Thus, only the component $v^{\top}(s)$ must be recovered to calculate $V(s)$. Nevertheless, a given set of $v^{\perp}(s)$ measurements along a contour is not enough to determine the motion uniquely. Therefore, in order to calculate the velocity $V(s)$ uniquely, one needs to integrate over the constraints provided by $v^{\perp}(s)$ along the contour combined with additional constraints, e.g. a smoothness constraint.
In order to find the velocity field that varies the least, Hildreth suggested three approaches to measured the variation in velocity along the contours.
First, he defined the local variation in $V(s)$ with respect to the contour given by $\partial V/\partial s$ (cf Fig. 3.11). Taking its magnitude, one obtains a scalar measure. A measure of the total variation in the velocity field over an entire contour may be derived by integrating this local measure, which leads to the following functional:

$$\Theta(V) = \int \left| \frac{\partial V}{\partial s} \right| ds.$$

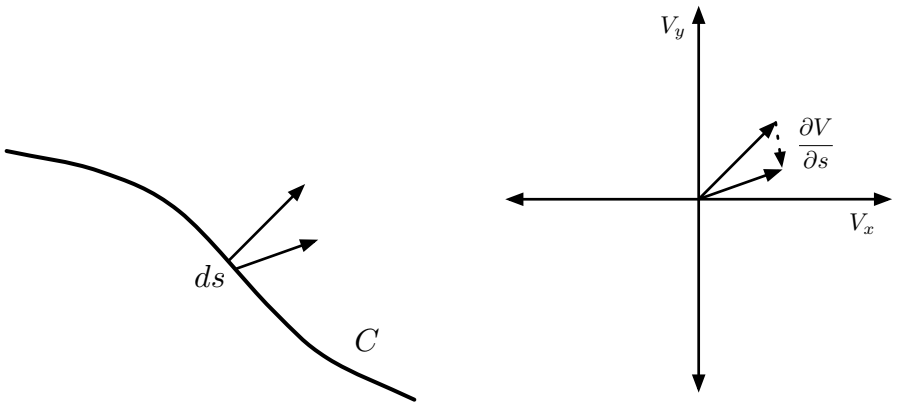
Second, he defined the variation in direction (cf Fig. 3.12), where the local change in direction for two nearby velocities is given by $\partial\varphi/\partial s$, where φ is the angle describing the direction of velocity (counterclockwise orientation). This leads to:

$$\Theta(V) = \int \left| \frac{\partial \varphi}{\partial s} \right| ds.$$

Finally, he suggested the total variation in magnitude of the velocity, which results in:

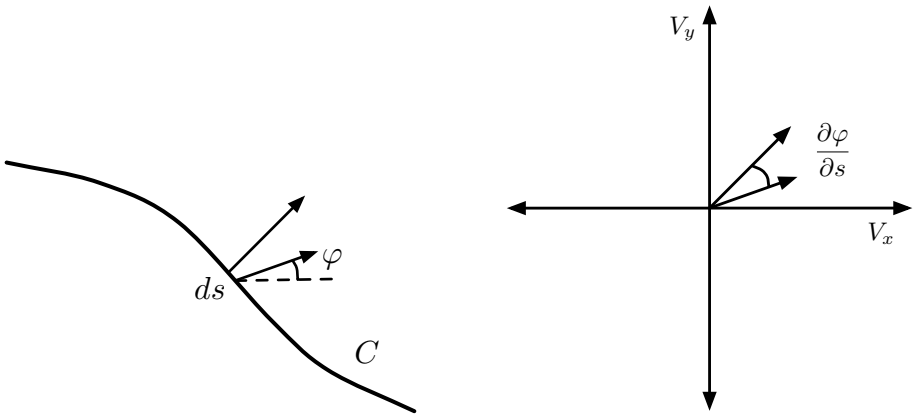
$$\Theta(V) = \int \frac{\partial |V|}{\partial s} ds.$$

He found out, that the use of functionals that incorporate only a measurement of direction or magnitude of velocity does not, in general, lead to a unique velocity field solution. However,



(a) Image space. (b) Velocity space.

Figure 3.11: Illustration of variation in $V(s)$. Figure (a) shows two nearby velocity vectors on the contour C. Figure (b) shows the according velocities in the velocity space, where $\partial V/\partial s$ is marked by the dashed line.



(a) Image space. (b) Velocity space.

Figure 3.12: Illustration of variation in direction. Figure (a) shows the direction of velocity represented by φ. Figure (b) shows the velocity vectors of the two nearby velocities of Figure (a) in velocity space. Here $\partial\varphi/\partial s$ indicates the change in direction.

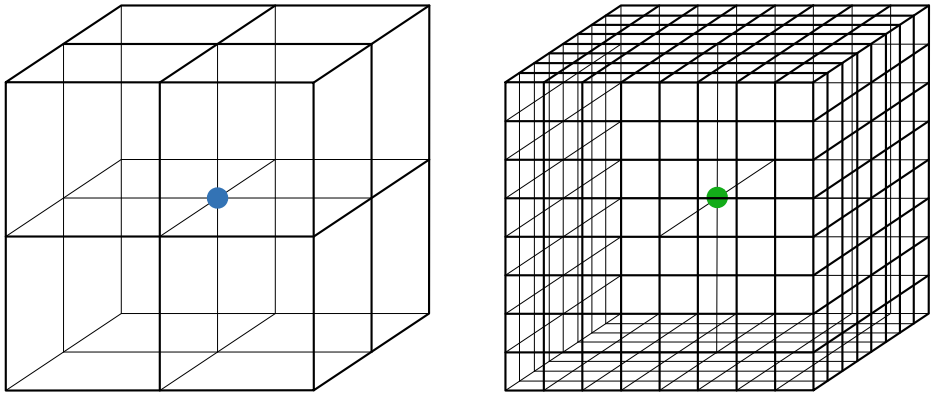
he showed, that using the measure of variation

$$\Theta(V) = \int \left| \frac{\partial V}{\partial s} \right|^2 ds \tag{3.33}$$

leads to a unique velocity field, that satisfies the known velocity constraints and minimizes (3.33), under the simple condition that $v^{\perp}(s)$ is known everywhere along the contour, and there exists at least two points at which the local orientation of the contour is different.

3.3.2.4 Gradient Estimation

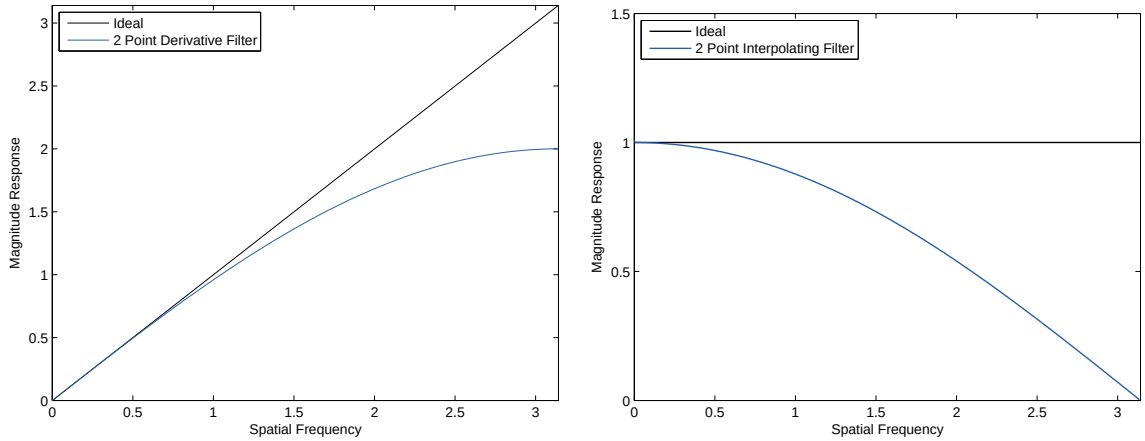
By considering the optical flow problem as a signal processing problem, it can be seen that the accuracy of the method is linked with choosing good derivative and interpolating filters. For the two image case, one can use e.g. two point derivative and interpolating filters like $[-1, 1]$, $[1/2, 1/2]$ respectively. These filters calculate the velocity of a point midway between two consecutive pixels in time and space (cf Fig. 3.13).



(a) $2 \times 2 \times 2$ pixel cube. (b) $N \times N \times N$ pixel cube.

Figure 3.13: Illustration of multi image gradient calculation using long filters. (a) shows the use of two point derivative and interpolating filters. Here, the velocity at the center of a $2 \times 2 \times 2$ pixel cube is calculated (marked by the blue dot). (b) shows the use of N point derivative and interpolating filters. Here the velocity is calculated at a $N \times N \times N$ pixel cube (marked by the green spot).

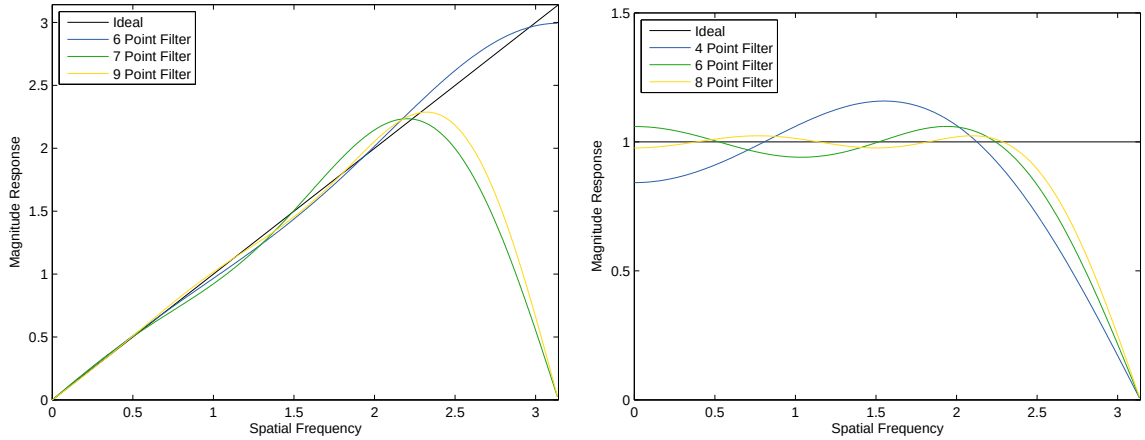
By using multiple images gathered over time, the accuracy of the calculated spatial and temporal derivative and interpolating filters can be increased by using long filters.
An ideal interpolating filter has a frequency response, whose magnitude equals one for all frequencies. An ideal derivative filter, meanwhile, has a magnitude response equal to the frequency itself. Figure 3.14 shows the response of the two point derivative filter $[-1, 1]$ and the two point interpolating filter $[1/2, 1/2]$, compared to their ideal counterparts. It can be seen that the two point filters significantly deviate from ideal behavior for high frequencies.



(a) Two point derivative filter.

(b) Two point interpolating filter.

Figure 3.14: Magnitude response of the two point derivative and interpolating filter. (a) shows the magnitude response of the two point derivative filter (blue), along with the ideal counterpart (black). (b) shows the magnitude response of the two point interpolating filter (blue) with the corresponding ideal response (black).



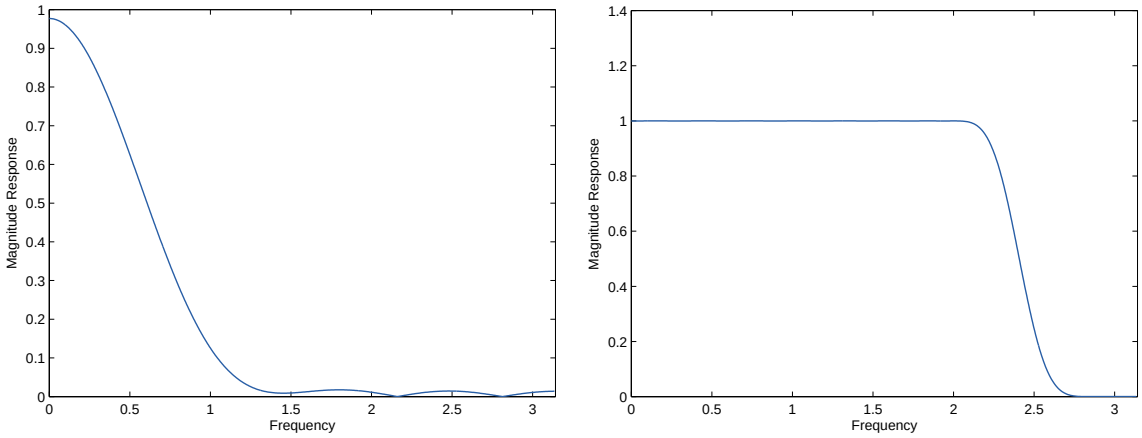
(a) Derivative filters.

(b) Interpolating filters.

Figure 3.15: Even and odd derivative and interpolating filters. The two figures show even and odd length derivative and interpolating filters, designed with the Parks-McClellan algorithm.

As mentioned above, the use of multiple images now allows the use of long filters. In Figure 3.15 the fundamental tradeoff between even and odd length filters is shown. Odd length derivative filters best approximate the derivatives for points at pixel centers. However, due to their symmetry, the magnitude response of odd length filters are 0 at frequency $\omega = \pi$. Even length filters best approximate derivatives for points, half way between pixel centers. Compared to odd length derivative filters the magnitude response of even length filters better approximates the ideal derivative filter, especially for high frequencies. However, the symmetry of the filter requires that the magnitude response of the interpolating filters of even length are 0 at $\omega = \pi$. Therefore even length derivative filters are better than odd length filters of compared length. On the other hand, odd length filters are better interpolating filters (always length 1) than even length filters.
Long derivative and interpolating filters can be calculated using the Parks-McClellan algorithm. Much more accurate derivative and interpolating filters can be designed by calculating them together rather than separately. This means that a small error in the derivative filter can be corrected by a corresponding error in the interpolating filter, and vice versa.

3.3.2.5 Prefiltering



(a) Gaussian low pass filter.

(b) Optimized low pass filter.

Figure 3.16: (a) shows the magnitude response of a Gaussian low pass filter. The filter response drops off very quickly, which results in the attenuation of the signal even at low frequencies. (b) shows the magnitude response of an optimized low pass filter. Here the filter response remains one for low frequencies and drops very fast at the cut off frequency.

To reduce the effects of high frequency noise many researchers use prefilters to low pass filter the images before calculating the optical flow. A popular prefilter is a Gaussian low pass filter, shown in Figure 3.16(a). The magnitude response of a typical Gaussian low pass filter decreases even for low frequencies. This has an undesired effect for images undergoing non-uniform motion. More precise, the movement of adjacent features by different amounts from

one image to the next has the effect, that the images will be 'stretched' in some areas and 'compressed' in others. These stretching and compressing effects represent changes in spatial frequencies of the images. Therefore, the spatial frequency of a target feature's neighborhood may slightly change from one image to the next. By applying a Gaussian low pass filter, the target feature's intensity will change too from one image to the next. This is because the Gaussian prefilter will attenuate image regions with higher frequencies (compressed neighborhoods) more, than image regions with the lower frequencies (stretched neighborhoods). This intensity variation violates the constant brightness assumption, and will reduce the accuracy of the calculations.
In order to avoid this effect, one has to use a prefilter, whose magnitude response remains constant for low frequencies, and drops off quickly at the cutoff frequency (cf Fig. 3.16(b)).

3.3.3 Matching Techniques

Matching techniques define velocity as the shift $\vec{d} = (d_x, d_y)^T$ that yields the best fit between regions at different times. The matching is performed by dividing the image into small regions (interrogation areas), which are then tracked from one image to the next by minimizing the following error function:

$$\sum_{(x,y)\in\Omega} \rho(I(x,y,t), I(x+d_x, y+d_y, t+1)), \tag{3.34}$$

where Ω is the interrogation area, (d_x, d_y) is the displacement vector, $I(x,y,t)$ is the image at spatial and temporal coordinates, and $\rho(.)$ is a function, computing the dissimilarity between two images. Common choices for $\rho(.)$ are the squared difference or the absolute difference between two arguments, leading to the following error functions:

$$\begin{aligned} \text{SSD:} &\sum_{(x,y)\in\Omega} (I(x,y,t) - I(x+d_x, y+d_y, t+1))^2, \\ \text{SAD:} &\sum_{(x,y)\in\Omega} |I(x,y,t) - I(x+d_x, y+d_y, t+1)|. \end{aligned} \tag{3.35}$$

The error for each interrogation area in the first image is calculated for a range of displacements in x and y direction in the second image. The displacement with the smallest error is chosen to be the true displacement. It should be mentioned that region-based matching can only calculate integer displacements. For a subpixel displacement calculation the images need to be interpolated.
Another evaluation function for region-based matching is the discrete normalized cross-correlation (NCC) function:

$$\text{NCC:} \frac{\sum_{(x,y)\in\Omega} \left(I(x,y,t) - \bar{I}_t\right) \left(I(x+d_x, y+d_y, t+1) - \bar{I}_{t+1}\right)}{\sqrt{\sum_{(x,y)\in\Omega} \left(I(x,y,t) - \bar{I}_t\right)^2 \sum_{(x,y)\in\Omega} \left(I(x+d_x, y+d_y, t+1) - \bar{I}_{t+1}\right)^2}}, \tag{3.36}$$

where $\bar{I}_t$ and $\bar{I}_{t+1}$ define the mean of $I(x, y, t)$ and $I(x + d_x, y + d_y, t + 1)$, respectively, for $(x, y) \in \Omega$. Unlike the sum of squared differences (SSD) or the sum of absolute differences (SAD) the cross-correlation function approaches one if there is a perfect match and goes to zero if there is no match. Therefore, the cross-correlation function is called to be a similarity measure. For the computation of the disparity the cross-correlation function is calculated for a range of displacements in x and y direction, which produces a cross-correlation table. The true displacement corresponds to the highest peak in this table.
Region-based matching can be computationally intensive. To speed up the computation of the cross-correlation table, the images are transformed to frequency domain to make use of the correlation theorem. Similar to the convolution theorem the correlation theorem relates spatial correlation to the product of the image transforms. If " $\circ$ " denotes a correlation operator and " $*$ " the complex conjugate, then the correlation theorem states the following:

$$f \circ w \Longleftrightarrow \mathcal{F}\{f\}\mathcal{F}^*\{w\}, \tag{3.37}$$

where f is an image and w is a given subimage (called mask or template) [13].

3.3.4 Spatio-temporal Filtering Techniques

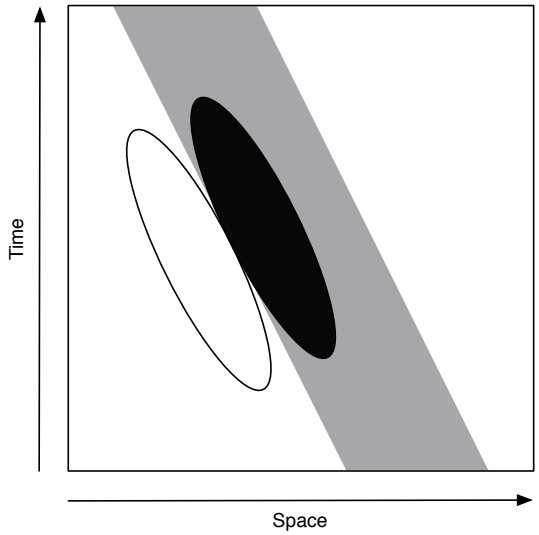


Figure 3.17: Space-time illustration of a leftward moving 1D signal. The movement of the signal produces a diagonal path in the 2D space-time diagram. The inverse slope of the bar corresponds to the speed of the 1D signal. Furthermore, the figure shows the positive and negative coefficient lobes of an idealized linear operator, where the orientation matches the underlying signal. [29]

Another method for motion analysis is based on spatio-temporal filters. Here, the basic motivation is that motion corresponds to orientation in space-time.
By way of illustration, we first consider a 1D example of a box signal. If one views this signal

in the space-time domain, as shown in Figure 3.17, it can be recognized that translation corresponds to a skewed bar. Furthermore, the inverse slope of this bar corresponds to the speed of the 1D signal. Thus, the motion can be determined by applying a set of oriented linear operators, where each responds best to signals that match its orientation. Figure 3.17 also shows the idealized impulse response lobes of a spatio-temporally oriented filter that matches the motion of the box signal.
Note, that the spatio-temporally filter response does not only depend on the velocity of the underlying signal, but also on symmetry, contrast, and spatial orientation of the underlying signal. Therefore, a linear filter response by itself does not constitute a velocity estimator [29].
To eliminate above mentioned dependencies, Adelson and Berg [1], e.g., suggested the computation of motion energy measures from the sum of the squares of even and odd-symmetric filters, tuned for the same orientation. Moreover, they suggested to subtract the output of a mechanism sensitive to leftward motion from one sensitive to rightward motion. They finally proposed a mechanism for computing a signal that is monotonically related to speed:

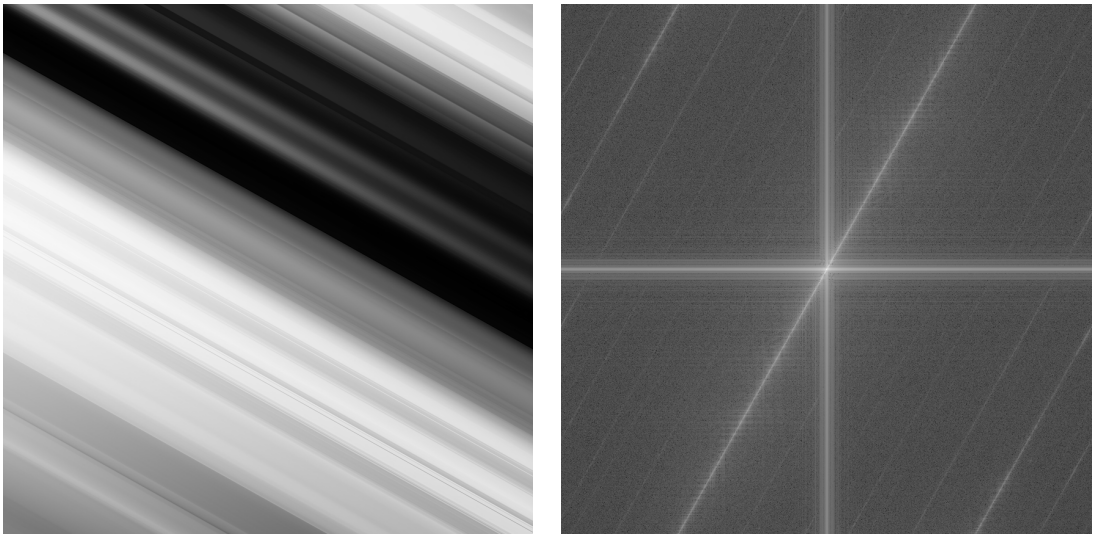
$$\frac{\sum(R_o^2 + R_e^2) - \sum(L_o^2 + L_e^2)}{\sum(S_o^2 + S_e^2)}. \tag{3.38}$$

R, L, and S here indicate the output of a filter tuned for rightward, leftward, and static motion, respectively. e and o refer to the even- and odd-symmetry of the filters.

3.3.5 Frequency-Based Techniques

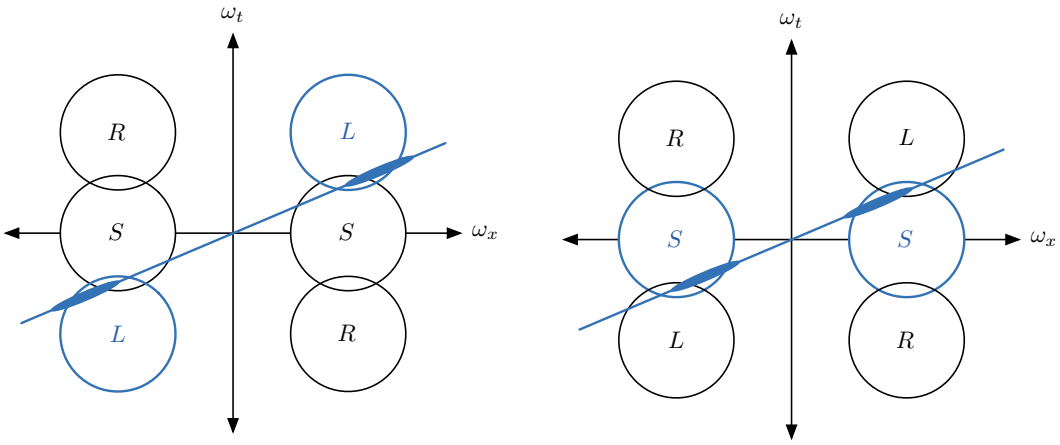
Frequency-based techniques are based on spatio-temporally oriented filters. They are motivated by considering the motion problem directly in the Fourier domain. These methods are divided into two main categories, namely energy based and phase based techniques.

Energy Based
First, the motion of a discrete 1D signal is considered. Such a signal can be represented by an intensity image, where each pixel intensity corresponds to the value of the signal at a particular location. By translating this signal it will appear as a striped pattern in the space-time domain, where the stripes are oriented at an angle $\alpha = arctan(1/u)$ (cf Fig. 3.18(a)). The corresponding Fourier decomposition is a set of sinusoids of the same orientation α, and varying wavenumbers. Therefore, the Fourier transformation will only have high values on a line through the origin at angle α (cf Fig. 3.18(b)).
This method can be extended straightforward to 2D. Thus, the Fourier transform spectrum of an image undergoing rigid transformation lies in a plane in the spatio-temporal frequency domain. This suggests an alternative approach to measure optical flow, by searching for the plane that best fits the power spectrum of the spatio-temporal signal.
Note that due to the fact, that we are interested in local estimates of the image velocity, we also need a local estimate of the power spectrum.
Heeger [17], e.g., used an energy based approach to develop an algorithm for computing optical



(a) Leftward moving 1D signal. (b) Fourier Spectrum.

Figure 3.18: Illustration of the main idea of frequency based techniques. Figure (a) shows the space-time diagram of a leftward moving 1D signal. Figure (b) shows the according Fourier spectrum, where the pattern lies on a line.



(a) Over-estimation of the signal-speed. (b) Under-estimation of the signal-speed.

Figure 3.19: Illustration of the systematic errors in the velocity estimation calculated with Gabor filter. The Gabor filter are represented by the overlapping circles and the power spectrum of the signal lies on the blue line. Figure (a) shows the situation, where the spectral distribution is concentrated at higher spatial frequencies (marked by the bulge). This increases the response of the leftward filter and leads to an over-estimation of the signal-speed. On the other hand, Figure (b) shows the situation, where spectral content at lower frequencies under-estimates the speed of the signal. [29]

flow. He first measured the power spectrum, using a set of Gabor filters [2], tuned for different spatio-temporal frequencies. In a second step he used a numerical optimization procedure to find the plane that best fits the measurements.
A drawback of this approach is that the resulting velocity estimates depend on local spatial content of the signal. Thus, this technique will calculate the wrong velocity for any sinusoidal grating, whose spatial frequency is not matched with the center response of the filter (cf Fig. 3.19).

Phase Based
Phase based methods initially decompose an image into band-pass channels. They assume a conservation of phase in each band-pass channel. The phase based gradient constraint for a given complex-valued band-pass channel $r(\vec{x}, t)$, with phase $\phi(\vec{x}, t) \equiv arg[r(\vec{x}, t)]$, is

$$\nabla\phi(\vec{x}, t) \cdot \vec{u} + \phi_t(\vec{x}, t) = 0. \tag{3.39}$$

Equation 3.39 can be used to estimate optical flow with any estimator described in Section 3.3.2.
Phase has some attractive properties for optical flow estimation. Phase is amplitude invariant, and thus quite stable in terms of contrast and intensity changes between images. Phase is also approximately linear over relatively large spatial extends and has very few critical points. This implies that, compared to image derivative approaches, more gradient constraints may be available, and that the range of velocities that can be estimated is significantly larger.
However, the main disadvantages of phase based methods are the computational costs of the band-pass filters, and the spatial support of the filters near occlusion boundaries and fine scaled objects.

3.3.6 Temporal Aliasing

Temporal aliasing is a typical problem of computing optical flow of real image sequences. A typical example is the wheel of a car, where the direction of rotation appears to reverse, when the wheel rotates at the right speed. This phenomenon is the result of the sparseness of the temporal sampling, introduced by the camera, and can cause problems for any type of motion algorithm. In the case of matching-based algorithm, this problem results in mismatchings. For filtering algorithms, the effect can be observed in the frequency domain. Therefore, one has to consider a 1D signal moving with constant velocity. As mentioned in the previous section, the power spectrum of this signal lies on a line through the origin. Temporal sampling introduces spectral replicas, causing aliasing for high speeds (cf Fig. 3.20) [29].
It is important to mention that temporal aliasing affects the higher spatial frequencies of an image. More precise, spatial frequencies that move more than half of their period per frame will be aliased, but lower spatial frequencies will not.
This gives rise to an approach for avoiding temporal aliasing by using a low frequency or coarse-scale prefilter that ignores higher frequencies. If the given imagery only contains a

[2] A Gabor function is a sinusoid multiplied by a Gaussian window.

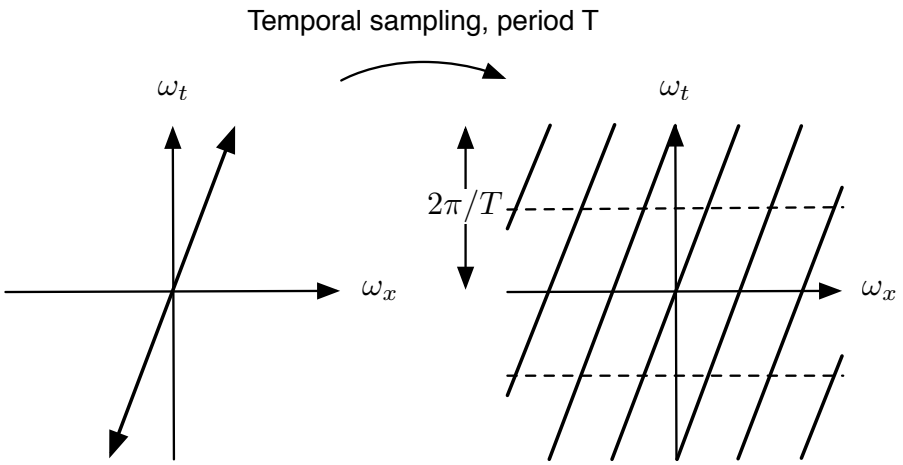


Figure 3.20: Illustration of temporal aliasing in the frequency domain. The left diagram shows the idealized power spectrum of a 1D pattern. Temporal sampling produces a replication of the power spectrum with temporal frequency intervals of $2\pi/T$, where T depicts the period of the signal. [29]

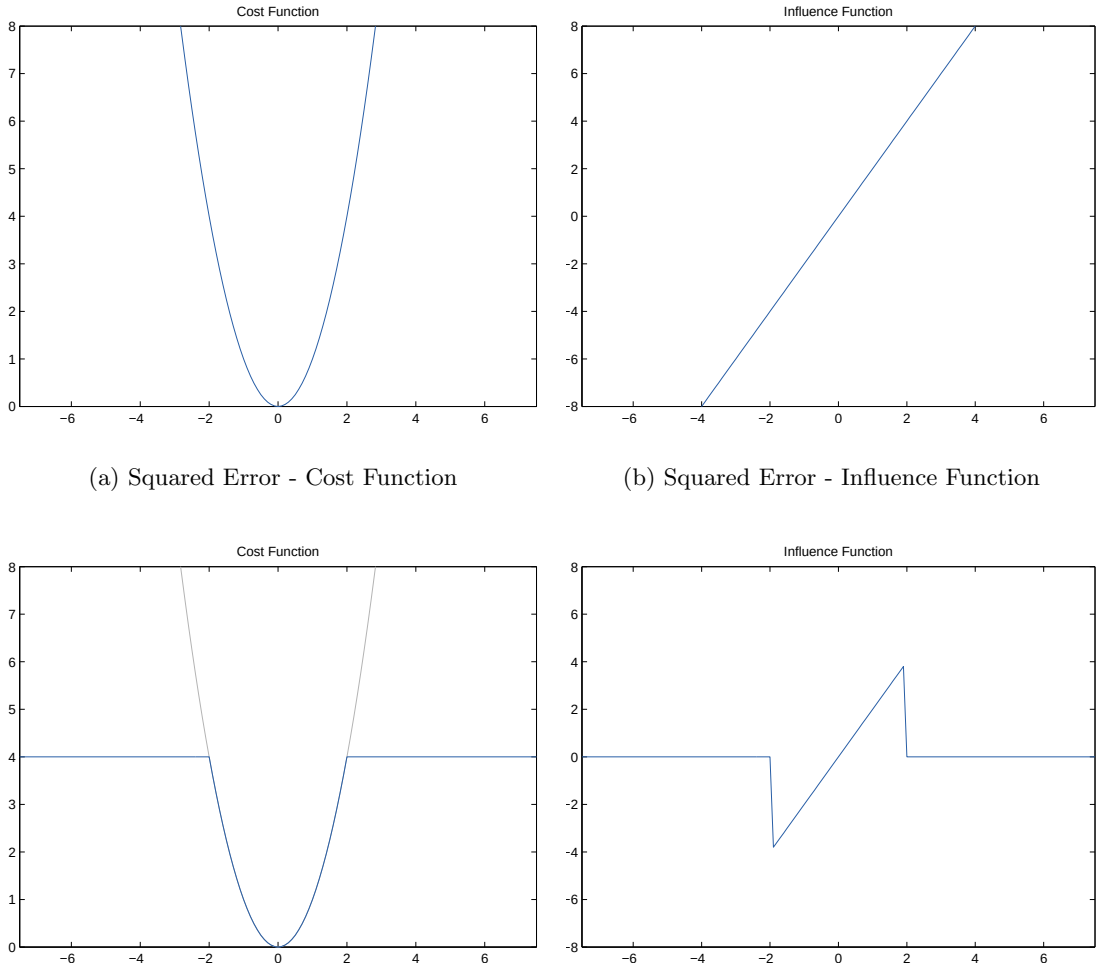
global motion, that varies slowly, we are done. However, in real-world scenes, an assumption of slowly varying motion fields is frequently violated. In order to receive better estimates of local velocity, higher frequency bands and spatially smaller filters must be taken into account. Therefore, first the coarse-motion estimate is used to undo the motion in a warping step. Next, higher frequency filters are used to extract the larger-scale motion, which gives a new optical flow estimate. This correction process can be repeated for finer and finer scales, and is referred to as coarse-to-fine approach.
However, this technique has serious drawbacks. If the coarse-scale estimates are incorrect, there will be no chance for the finer-scale estimates to correct the errors. Thus, a poor estimate at one scale provides a poor initial guess at the next finer scale.

3.4 Robust Estimation

The two main goals of robust statistics according to Hampel [14] are, first, to describe the structure, best fitting the bulk of data, and second, to identify deviating data points (outliers) or deviating substructures for further treatment, if desired.
Robust estimation deals with the problem of finding parameters $a = [a_0, ..., a_n]$, that best fit a given model $m(s, a)$ to a set of data points $d = [d_0, ..., d_s]$, $s \in S$, where the data differs statistically from the model. The goal is to find values for a that minimize the residual errors

$$\min_a \sum_{s \in S} C(d_s - m(s, a), \sigma_s), \tag{3.40}$$

where $C(.)$ is an estimator and σ_s is a scaling parameter. When considering errors $\delta_s = d_s - m(s, a)$ that are mean-zero Gaussian, and independent and identically distributed (IID),



(a) Squared Error - Cost Function

(b) Squared Error - Influence Function

(c) Truncated Quadratic - Cost Function

(d) Truncated Quadratic - Influence Function

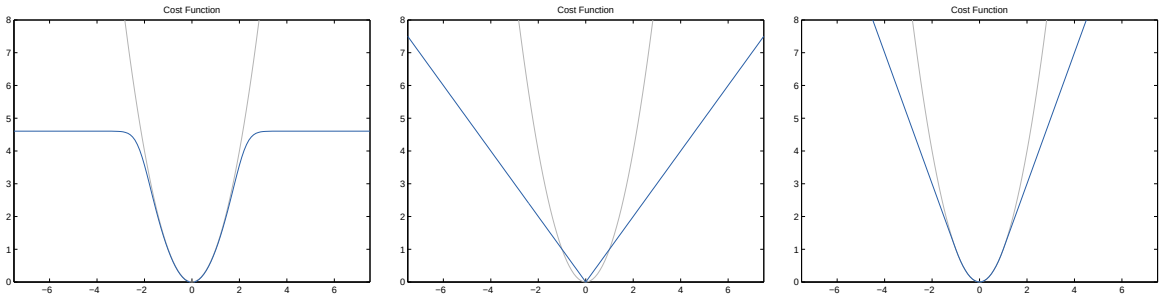
Figure 3.21: The first row shows the cost and influence function of the squared error. Here outliers are assigned a high weight, which can have negative effects on the estimation. The second row shows the cost and influence function of the truncated quadratic estimator. This estimator weights the errors quadratic up to a fixed threshold. Errors beyond that threshold receive a constant weight. As a result the influence of outliers goes to zeros, which makes the estimator more robust.

the optimal estimator is the quadratic (see Figure 3.21)

$$C(\delta_s, \sigma_s) = \frac{\delta_s^2}{2\sigma_s^2} \tag{3.41}$$

which leads to the standard least-squares estimation problem.
The robustness of an estimator is related to its insensitivity to outliers. Considering the quadratic estimator, we recognize that outliers are assigned a high weight, which can be seen by the according influence function. This function characterizes the bias that a measurement

has on the solution and is obtained by the deviation of the estimator. In the least-squares case, this influence function increases linearly without bounding.
In order to increase robustness one has to consider estimators for which the influence of outliers tends to zero. One such robust estimator is the truncated quadratic (cf Fig. 3.21). This estimator weights the errors quadratically, but only up to a fixed scale. Beyond that, errors receive a constant value. Therefore, the influence of outliers goes to zero beyond the threshold.



(a) Blake Zisserman (b) L1 (c) Huber

Figure 3.22: The Figures show, from left to right, the Blake-Zisserman, the L1, and the Huber cost-function (represented by the blue functions) compared to the quadratic estimator (represented by the gray function).

Figure 3.22 shows, from left to right, the Blake Zisserman, the L1, and the Huber cost function, which will be briefly discussed below [15].

Blake-Zisserman Cost Function
The Blake-Zisserman cost function (see Figure 3.22(a)) is considered as a statistically based cost function. This means that the cost function is derived by guessing the distribution of errors for the particular measurements. Therefore, the probability density function (PDF) is assumed to be $\rho(\delta) = exp(-\delta^2) + \epsilon$, which is in fact not actually a PDF, but leads to the following cost-function:

$$C(\delta) = -log(exp(-\delta^2) + \epsilon). \tag{3.42}$$

Note, that the normalization constant for the Gaussian distribution $1/\sqrt{2\pi\sigma^2}$ is ignored. Furthermore, it is assumed that $2\sigma^2 = 1$. This cost function weights inliers (small δ) quadratic, and outliers (large δ) are assigned a cost of $-log(\epsilon)$. Note, that this is a non-convex cost function, which can lead to multiple local minima.

L1 Cost Function
The L1 cost function (cf Fig. 3.22(b)) uses the sum of absolute errors instead of the sum of squares. Therefore, the cost-function is

$$C(\delta) = 2b|\delta|, \tag{3.43}$$

where $2b$ is a positive constant. Compared to the quadratic cost function, outliers are less weighted here. Note that this cost-function is convex, which leads to a single minimum, but it is non-differentiable at the origin.

Huber Cost Function
The last cost-function considered here is the Huber cost-function (cf Fig. 3.22(c)). Its is a hybrid between the L1 and the quadratic cost-function. Thus,

$$C(\delta) = \begin{cases} \delta^2, & \text{for } |\delta| < b, \\ 2b|\delta| - b^2, & \text{otherwise,} \end{cases} \tag{3.44}$$

where b defines the outlier threshold. This cost function is continuous with a continuous first derivative, and also convex.

3.5 Theory of Defocus

To calculate the depth information, using a wavefront sampling approach, one first has to know how a target feature's depth is encoded by the diameter of its defocus blur spot on the image plane. Figure 3.23 shows the sketch of an imaging system, where a point at distance

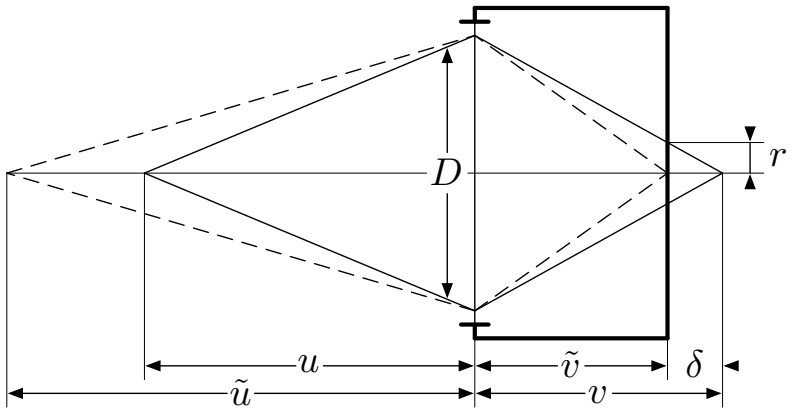


Figure 3.23: Sketch of an imaging system with aperture D. A point at distance $\tilde{u}$ in front of the lens is viewed in focus, where a point at distance u is blurred on the image plane with radius r. [28]

$\tilde{u}$ in front of the lens produces a in-focus image on the sensor at distance $\tilde{v}$ behind the lens. As the target feature moves away from the in-focus plane, the diameter of the defocus blur increases. Hence, a target at distance u produces a defocused image, with a blur-circle of radius r on the image plane. The blur radius r can be calculated using a straightforward

geometrical optics analysis [8]. Considering Figure 3.23, we recognize that similar triangles yield a formula for the radius r of the blur circle

$$\frac{2r}{\delta} = \frac{D}{v} \Longrightarrow \frac{2r}{D} = \frac{\delta}{v}, \tag{3.45}$$

where D is the diameter of the aperture, v is the distance from the lens to a focused image of an object at distance u in front of the lens, and δ is the displacement of the image plane from sharp focus. Thus,

$$\delta = v - \tilde{v}. \tag{3.46}$$

By substituting Equation 3.46 into Equation 3.45, and by applying the Gaussian lens law

$$\frac{1}{v} + \frac{1}{u} = \frac{1}{F} = \frac{1}{\tilde{v}} + \frac{1}{\tilde{u}}, \tag{3.47}$$

one gets:

$$\frac{2r}{D} = \frac{v - \tilde{v}}{v} = 1 - \frac{\tilde{v}}{v} = \tilde{v}\left(\frac{1}{\tilde{v}} - \frac{1}{v}\right) = \tilde{v}\left(\left(\frac{1}{F} - \frac{1}{\tilde{u}}\right) - \left(\frac{1}{F} - \frac{1}{u}\right)\right) = \tilde{v}\left(\frac{1}{u} - \frac{1}{\tilde{u}}\right),$$

where F is the focal length. Thus, an expression that directly relates the radius r or rather the diameter $d = 2r$ of a target feature's defocus blur circle to the feature's distance from the lens u is obtained:

$$\frac{d}{D} = \tilde{v}\left(\frac{1}{u} - \frac{1}{\tilde{u}}\right). \tag{3.48}$$

Evaluation of Equation 3.48 over a range of normalized target depths results in the plot shown in Figure 3.24. It can be seen, that the blur diameter equals zero, when the target is located on the in-focus plane. The blur diameter increases when the target moves away from the in-focus plane. Therefore, the accuracy of this method is directly related to the slope of the line in Figure 3.24. Moreover, Figure 3.24 shows that the depth sensitivity increases more steeply, if the target moves towards the lens, than away from the lens. This shows that the optimal operating distance for a depth calculating system based on the wavefront sampling approach is defined somewhere between the in-focus plane and the lens.
It should also be mentioned that for a given disparity measurement there are two possible target positions. The first one is located between the lens and the in-focus plane. The second position is located beyond the in-focus plane. Thus, without any prior knowledge this leads to a depth ambiguity.

3.5.1 Defocus Measure

An image $i(\vec{x})$, captured by an image sensor, is modeled by the convolution of a sharp pre-image $s(\vec{x})$ with the point spread function (PSF) $h(\vec{x})$

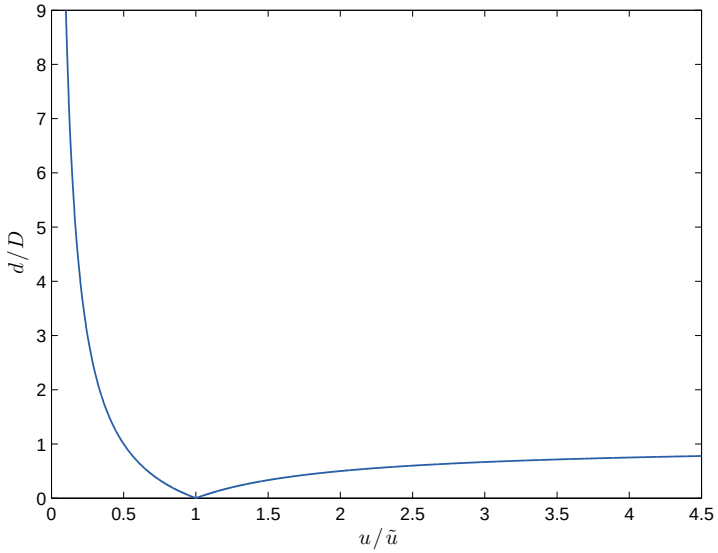


Figure 3.24: Normalized target distance as a function of the normalized blur spot diameter. Here, d is the diameter of the blur spot, D is the diameter of the lens' exit pupil, u is the target distance to the lens, and $\tilde{u}$ is the distance between lens and in-focus plane.

$$i(\vec{x}) = s(\vec{x}) * h(\vec{x}), \tag{3.49}$$

where $\vec{x} = (x, y)^T$ are the spatial coordinates.
The PSF or impulse response of an optical system is the irradiance distribution that results from a single point in object space. Although the source is a point, the image may not. The two main reasons for the spread over a finite area are aberrations in the optical system, and diffraction effects. Thus, the PSF $h(\vec{x})$ can be considered as a convolution of the optical kernel $\mu(\vec{x})$, the defocus kernel $\eta(\vec{x})$, and the sampling blur kernel $\rho(\vec{x})$ [24]:

$$h(\vec{x}) = \int\int \rho(\vec{x} - s)\eta(s - t)\mu(t)dsdt \tag{3.50}$$

The diffraction blur $\mu(\vec{x})$ in Equation 3.50 is given by

$$\mu(x, y) = \frac{2J_1(\gamma)}{\gamma}, \text{ with } \gamma = \frac{\pi\|\vec{x}\|D}{\lambda v}. \tag{3.51}$$

J_1 in above equation is the first-order Bessel function, and λ is the wavelength of light ($= 0.7 \times 10^{-6} m$). The defocus kernel has a cylindrical shape and is given by the following pillbox function with radius r:

$$\eta(x, y) = \begin{cases} \frac{1}{\pi r^2}, & \|\vec{x}\| \leq r^2 \\ 0, & \text{otherwise.} \end{cases} \tag{3.52}$$

The sampling kernel describes averaging over a square pixel of size Δx:

$$\rho(x,y) = \begin{cases} \frac{1}{\Delta x^2}, & \text{if } max(|x|,|y|)) \leq \frac{\Delta x^2}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{3.53}$$

As mentioned above, blurring due to defocus can be modeled as a convolution with the PSF $h(x,y)$. But, a single unfocused image does not contain enough information in order to extract the defocus operator. Hence, at least two images of the same scene with different defocus operators are required. Thus, we need to change the defocus operator, which can be done by varying the position of the image plane $\tilde{v}$, the focal length F, or the aperture D.
A common method to isolate the defocus operator from the scene is inverse filtering, which will be described in Section 3.5.2.

3.5.2 Inverse Filtering

Ens and Lawrence [8] calculate the depth information from two defocused images, acquired with two different f-numbers[3]. Defocusing by change of the f-number preserves v (compare Figure 3.23). This has the advantage that unwanted scaling of the image is excluded. The blur is treated as a convolution of a sharp image $s(x,y)$ with a low-pass filter, similar to $\eta(x,y)$. First, a less blurred image

$$i_1(x,y) = s(x,y) * \eta_1(x,y) \tag{3.54}$$

is acquired with a defocused system with a f-number f_1. Next, a more blurred image

$$i_2(x,y) = s(x,y) * \eta_2(x,y) \tag{3.55}$$

with a f-number $f_2 < f_1$ is acquired. Afterwards, the inverse problem is solved to find a function η_3, which transforms $i_1(x,y)$ into $i_2(x,y)$:

$$i_1(x,y) * \eta_3(x,y) = i_2(x,y). \tag{3.56}$$

By substituting Equation 3.54 and 3.55 into Equation 3.56 one obtains

$$s(x,y) * \eta_1(x,y) * \eta_3(x,y) = s(x,y) * \eta_2(x,y) \tag{3.57}$$

$$\eta_1(x,y) * \eta_3(x,y) = \eta_2(x,y) \tag{3.58}$$

Thus, $\eta_3(x,y)$ is also called the convolution ratio of $\eta_2(x,y)$ and $\eta_1(x,y)$. Transformed into the Fourier domain, this deconvolution problem translates into a simple division defined by

$$\mathcal{F}\{\eta_3(x,y)\} = \frac{\mathcal{F}\{i_2(x,y)\}}{\mathcal{F}\{i_1(x,y)\}}, \tag{3.59}$$

which can be seen by considering Equation 3.54, 3.55, and 3.58. Thus, the function $\eta_3(x,y)$ has been isolated and the one-to-one relationship between η_3 and the depth is derived from geometric optics or found from a look-up table, evaluated on a calibration object.

[3]The f-number $f/\#$ is given by f/D, where f is the focal length, and D is the aperture diameter.

3.6 Wavefront Sampling

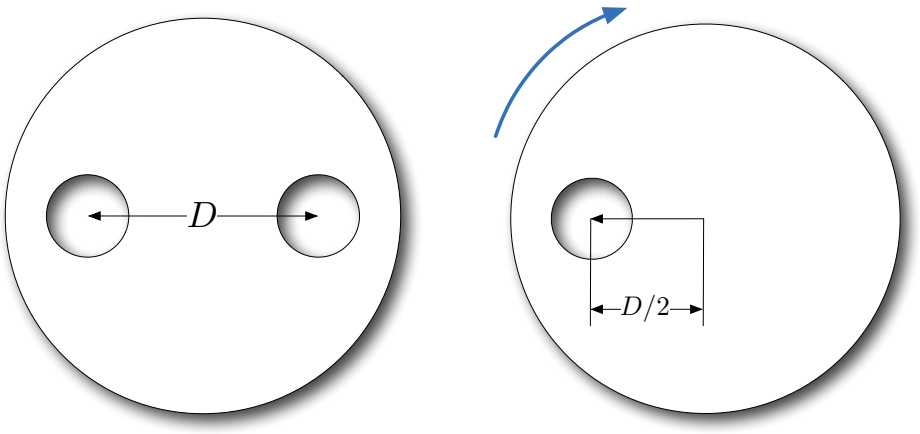
In a typical DFD system the whole wavefront is allowed to reach the image plane. As mentioned in the previous section, the challenging part is the accurate calculation of the diameter of each target feature's blur spot. Furthermore, the usability of a DFD system is usually limited due to complications occurring with overlapping blur spots from feature rich targets. By using a wavefront sampling approach some limitations in the DFD methods can be overcome.

3.6.1 Static Wavefront Sampling

Compared to a DFD system, the wavefront sampling approach only allows specific parts of the optical path's wavefront to travel through and hit the image plane. The sampling is done by using special sampling patterns. The simplest pattern is one with two apertures. The corresponding reconstruction method is referred to as Static Wavefront Sampling (SWS). Similar to the imaging system sketched in Figure 3.29, SWS results in the projection of two quasi-in-focus images of the target on the image plane. Assuming that the two apertures are separated by a distance equal to the full exit pupil diameter of the DFD system, the distance between the two quasi-in-focus images resulting from the SWS method will be exactly equal to the blur spot diameter resulting from the DFD technique. Hence, wavefront sampling can also be compared with sampling the defocus blur.
In order to determine a target feature's depth, the distance between the two images of the target feature needs to be calculated. This can be done by using any kind of matching technique. However, the fact that two images of the same target are recorded on the same image can still cause overlapping problems due to feature rich targets.

3.6.2 Active Wavefront Sampling

There are two main disadvantages concerning SWS methods. The first one is depth ambiguity, which means that it is not distinguishable whether the target is located in front or behind the in-focus plane. The second drawback is the possible overlap on feature rich targets.
The Active Wavefront Sampling (AWS) approach [10] provides a solution to both problems. In an AWS system a single aperture is moved from one position to the next. At each position a single image is recorded without multiple image overlap. Furthermore, the depth ambiguity is resolved as long as the motion of the aperture on the sampling plane is known. For instance, a single off-axis aperture, rotated in a circle, and centered on the optical axis of the lens, would have the effect that the target's image would also rotate in a circle. Hence, depth information for the target is encoded in the diameter of the rotation on the image plane. A target located on the in-focus plane will have a zero diameter rotation and thus will remain constant, whereas targets located at increasing distances from the in-focus plane will rotate along circles with increasing diameter. Note, that a target, located beyond the in-focus plane, will rotate 180° out of phase, compared to a target, located between the lens and the in-focus plane. As a result of this phase difference the depth ambiguity presented in the DFD and SWS systems can be solved.
It should be mentioned that a rotating off-axis aperture is not the only possible path that can



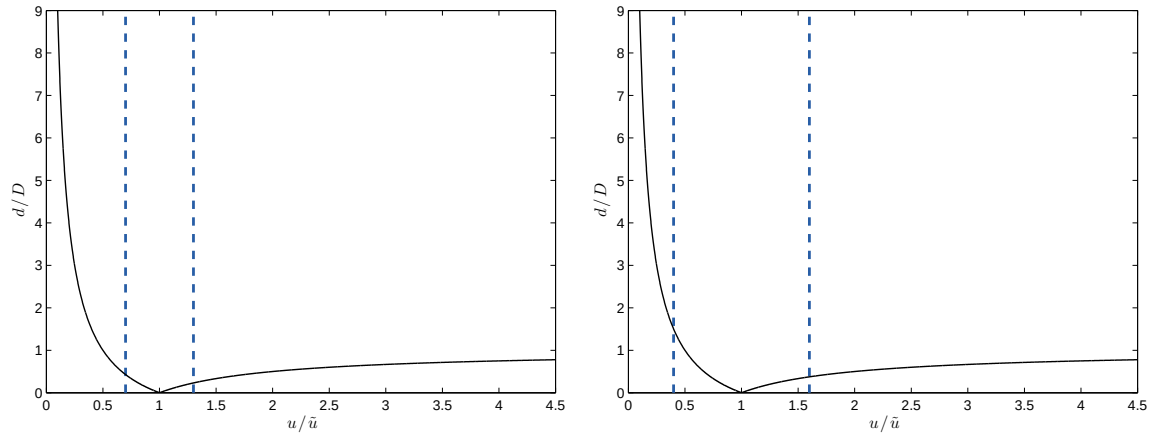
(a) SWS (b) AWS

Figure 3.25: (a) shows the sampling mask of a two aperture Static Wavefront Sampling (SWS) system. The wavefront is sampled by two diametrically opposed apertures separated by a diameter D. The use of such a sampling mask results in two quasi-focused images recorded on the image plane. (b) shows the sampling mask of an Active Wavefront Sampling (AWS) system. The wavefront is sampled by an off-axis aperture, that rotates around the optical axis. At each sampling position a quasi-focused image is acquired. These images are used to calculate the blur-spot diameters.

be used with AWS. Other possibilities are, e.g. simple translations along horizontal, vertical, or diagonal lines, or in more general, any path following any arbitrary closed loop. In theory, depth can be recovered as long as the aperture path is known. A simple circular aperture path has certain advantages, e.g. the relatively simple mechanical implementation.
One main advantage of the AWS method is the possibility to adjust the system with respect to accuracy and processing speed. This means, that for high speed imaging applications where some measurement accuracy can be sacrificed, the sampling positions can be reduced to a minimum of two. Otherwise, considering high accuracy applications where speed can be sacrificed, a high number of sampling positions can be used for the calculation.

3.6.3 Size of the Sampling Aperture

The size of the sampling aperture can strongly affect the quality of the image [10]. A large aperture will allow to image higher frequency features, but simultaneously it will also have a small depth of field, which reduces the depth range of the system to a narrower band, surrounding the in-focus plane (cf Fig. 3.26). It also allows shorter image exposure times, which increases the maximum frame rate of the imaging system.
On the other hand, a smaller aperture increases the depth of field, which results in a higher depth sensitivity, but it will also low pass filter the images. Furthermore, it will also increase



(a) Large aperture depth range.

(b) Small aperture depth range.

Figure 3.26: (a) Practical target depth range for a large aperture size. (b) Practical target depth range for a small aperture. [10]

the exposure time, which reduces the maximum frame rate of the system.

3.6.4 Placement of the Sampling Plane

Frigerio [10] showed that the aperture sampling plane should be placed as close as possible to the lens in order to minimize vignetting and intensity variation between consecutive images. The effect of placing the sampling aperture far from the lens is sketched in Figure 3.27, which shows that for the top and bottom position of the aperture point 1 and point 2 are blocked respectively. This vignetting effect has the result that a target feature image cannot be tracked from one sampling position to the next. Thus a loss of depth information occurs. Conversely, Figure 3.28 shows the sketch of an imaging system where the sampling plane is located close to the lens. Here we see that the vignetting effect is removed, which makes a full 3D reconstruction possible.
Another possible position for the sampling plane is at the lens's aperture plane, which will also minimize vignetting and intensity variation between images.

3.6.5 Comparison to Stereo Imaging

Now we consider Figure 3.29, where the entire lens is blocked, except for two pinholes on its perimeter, on opposite ends. As a result the geometrical point spread function consists of only two points, x_L and x_R, with distance

$$d = 2r = |x_R - x_L|. \tag{3.60}$$

Due to of the fact that the image of the target now consists of two separated points, it seems natural to compare it with a stereo setting. Therefore, we now consider the canonical

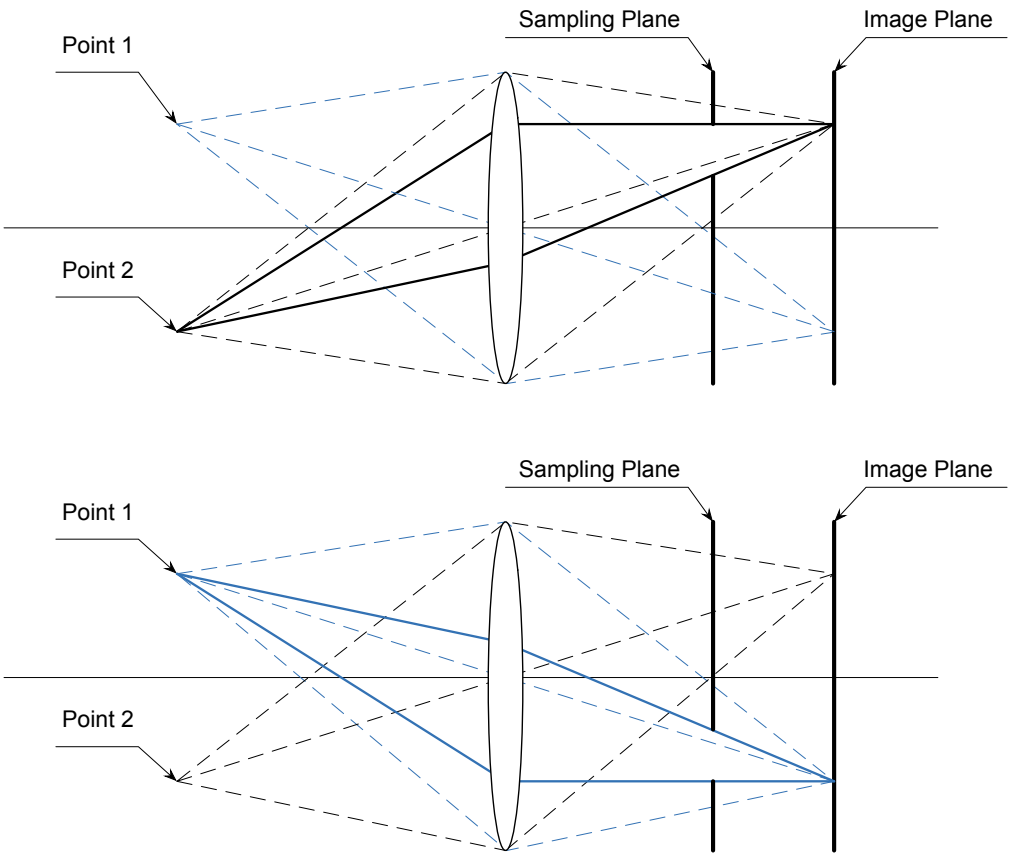


Figure 3.27: Effect caused by placing the sampling plane far away from the lens. The upper sketch shows the aperture located at the top. In this case light rays from point 2 are able to reach the image plane, but all rays from point 1 are blocked. The lower sketch shows the case, where the aperture is located on the bottom. Here rays from point 1 are able to reach the image plane, but rays from point 2 are blocked. This effect prevents a 3D reconstruction. [10]

stereoscopic system, sketched in Figure 3.30. This system consists of two pinhole cameras with the same physical dimensions of the system defined in Figure 3.23. The target point at distance u is again imaged to two points, one at each sensor. The disparity $d = \hat{x}_L + \hat{x}_R$ between these images is related to the target feature's distance to the cameras, as defined by:

$$\frac{d}{D} = \tilde{v} \cdot \frac{1}{u}. \tag{3.61}$$

Comparing Equation 3.61 with Equation 3.48, we recognize that they only differ by the constant term $1/\tilde{u}$.

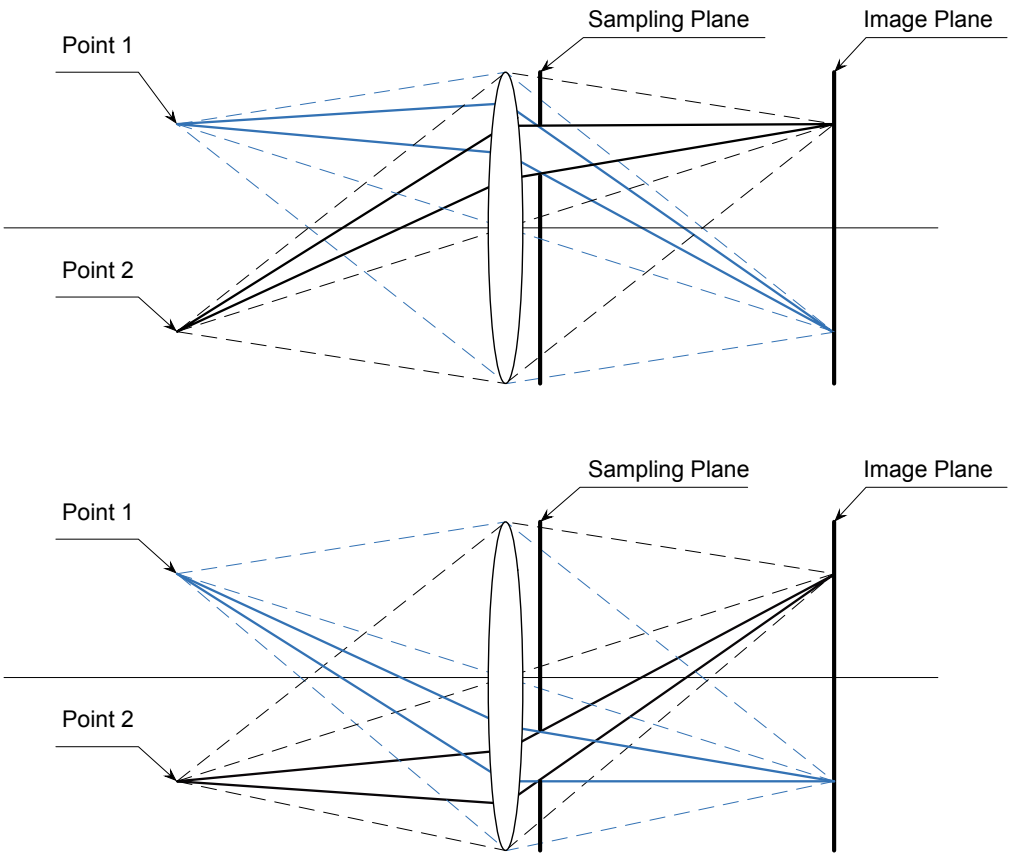


Figure 3.28: Effect caused by placing the sampling plane close to the lens. The upper sketch shows the aperture located at the top and the lower sketch shows the aperture located at the bottom. In both cases light rays from point 1 as well as from point 2 are able to reach the image plane. Thus a 3D reconstruction is possible. [10]

3.6.6 Depth Sensitivity of AWS based Systems

As mentioned above, the optimal operating regime of a DFD system is defined somewhere between the in-focus plane and the lens, as shown in Figure 3.24. Because of the similarities between AWS and DFD the idealized optical performance characteristics of these two approaches are identical. Thus the sensitivity can be calculated by taking the derivative of Equation 3.48 with respect to the target depth u [10]. This leads to

$$\begin{aligned} \frac{\partial}{\partial u}\frac{d}{D} &= \frac{\partial}{\partial u}\tilde{v}\left(\frac{1}{u}-\frac{1}{\tilde{u}}\right) \\ \frac{\partial d}{\partial u} &= -D\cdot\tilde{v}\cdot\frac{1}{u^2}. \end{aligned} \tag{3.62}$$

Now one can compare this with a canonical stereoscopic system with a disparity D between the two images (see Equation 3.61). Just like above, the sensitivity can be obtained by taking the derivative of this equation with respect to target depth, which leads to

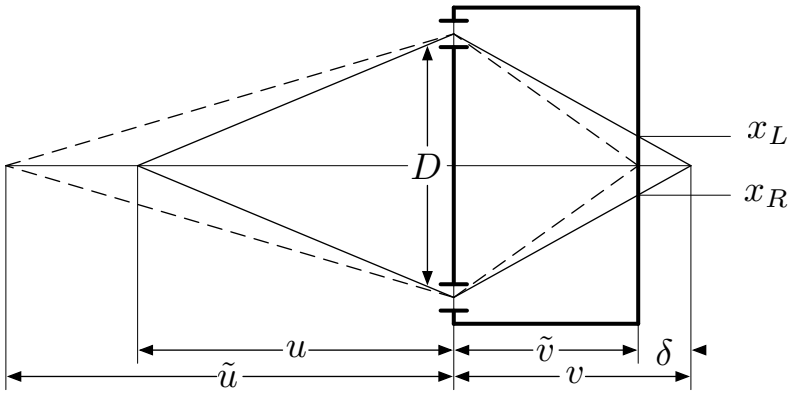


Figure 3.29: The sketch of an imaging system similar to that of Figure 3.23, but with the difference that the lens is now blocked except for two pinholes on its perimeter, on opposite ends. Therefore, an out-of-focus point at distance u produces two projections on the image plane, with a disparity equal to the diameter of the blur circle, that would have been appeared without a blocked lens. [28]

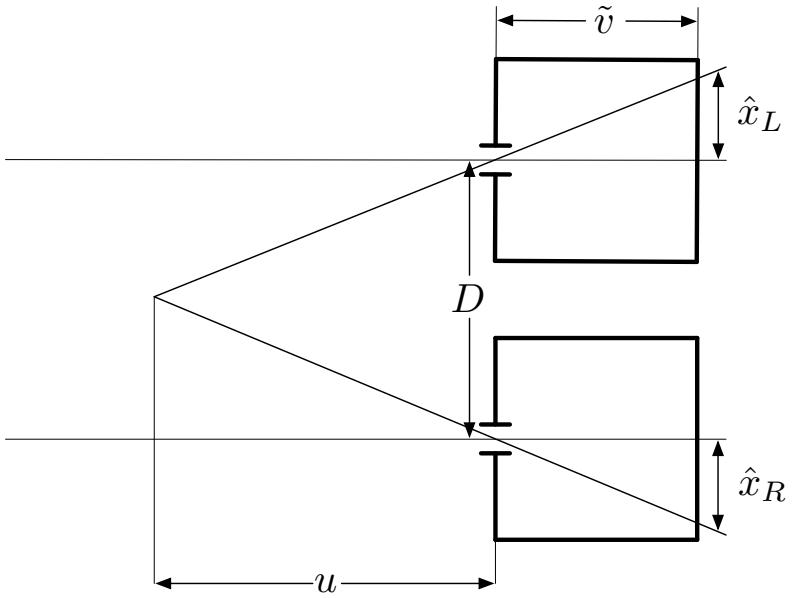


Figure 3.30: The sketch of a canonical stereoscopic system with baseline D. A point at distance u is imaged to two points, one at each sensor.

$$\frac{\partial}{\partial u}\frac{d}{D} = \frac{\partial}{\partial u}\tilde{v}\frac{1}{u}$$
$$\frac{\partial d}{\partial u} = -D \cdot \tilde{v} \cdot \frac{1}{u^2}. \tag{3.63}$$

By comparing Equation 3.62 and 3.63 it can be seen that the depth sensitivities of a canonical stereo system and an AWS system are identical. Hence, the only physical parameters that can be changed in order to increase the sensitivity to depth are the sampling diameter, which is the baseline in the stereo system, and the distance between the lens and the imaging sensor. Depth sensitivity increases with $\tilde{v}$. Simultaneously the field of view will decrease. The sensitivity to depth also increases with D, but in the case of AWS, the diameter D is limited by the lens exit pupil size. Thus, the bigger the lens, the larger the maximum sampling diameter.

3.6.7 Frigerio's Multi Image AWS Algorithm

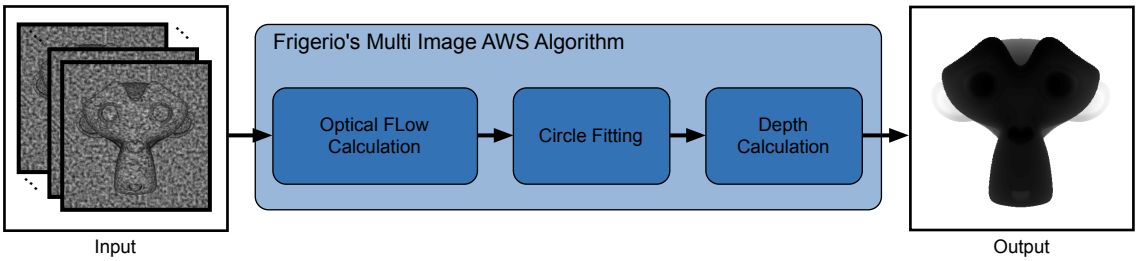


Figure 3.31: Frigerio's multi image AWS algorithm procedure. The figure shows the three main steps of Frigerio's AWS approach. The algorithm takes as input the images acquired by moving the rotating off-axis aperture from one position to the next. The output of the algorithm is the according depth map.

Frigerio [10] was the first, who presented an algorithm, that uses more than two AWS sampling positions. His multi image AWS approach consists of three steps (cf Fig. 3.31). In the first step, Frigerio defines an anchor image (located at the 9 o'clock position on the sampling plane). By rotating the aperture, new images are acquired and the displacements between the anchor image and the new acquired images are calculated. The displacement calculation is done in two steps: First the integer pixel displacements are calculated and then, after warping the images by the integer displacements, the remaining subpixel displacements are calculated using a gradient based optical flow approach.
In the second step, a circle is fitted to the points, which are calculated by adding the displacements to the according anchor image position. This is done in a least squares sense and by knowing the angular between the sampling positions. As a result Frigerio receives the rotation diameter of each pixel in the anchor image.
In a final step, these rotation diameters are used to calculate the depth of each pixel in the anchor image. This is done by using simple geometric considerations.
Step one and three are described in Section 3.3 and Section 3.5, respectively. Therefore, the

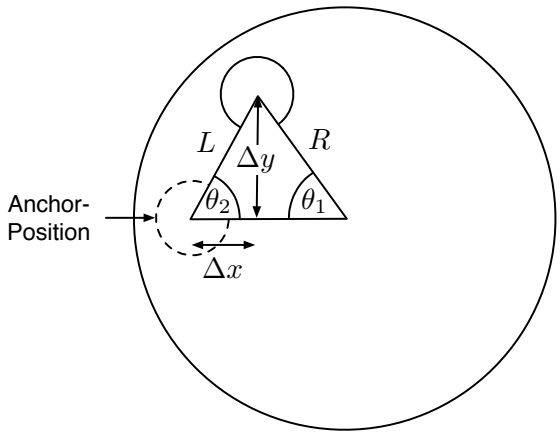


Figure 3.32: Sketch showing the geometry underlying the calculation of the rotation radius R.

only thing left to show is the circle approximation step. For this purpose, Frigerio first defined the angular θ_2 for each image pair. This angular θ_2 describes the motion direction of that image pair.

$$\theta_2 = tan^{-1}\left(\frac{\Delta y}{\Delta x}\right), \tag{3.64}$$

where $(\Delta x, \Delta y)$ is the x and y component of the motion. Next, Frigerio used the cosine law to calculate the motion radius R in terms of the motion distance L and the angle θ_1 separating the two images (cf Fig. 3.32).

$$L^2 = 2R^2 - 2R^2 cos\theta_1, \tag{3.65}$$

$$R = \frac{L}{\sqrt{2(1 - cos\theta_1)}}. \tag{3.66}$$

The motion distance L can also be calculated using Δx, Δy and θ_2 as follows:

$$L = \frac{\Delta x}{cos\theta_2}, \tag{3.67}$$

$$L = \frac{\Delta y}{sin\theta_2}. \tag{3.68}$$

By substituting Equations 3.67 and 3.68 into Equation 3.66, one obtains an overdetermined set of equations, which can be used to solve for the radius R.

$$\begin{pmatrix} \sqrt{2(1 - cos\theta_1)}cos\theta_2 \\ \sqrt{2(1 - cos\theta_1)}sin\theta_2 \end{pmatrix} R = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \tag{3.69}$$

By using N sampling positions, $N - 1$ image pairs can be processed and furthermore an

overdetermined system of equations with $2(N-1)$ equations can be used to solve for R.

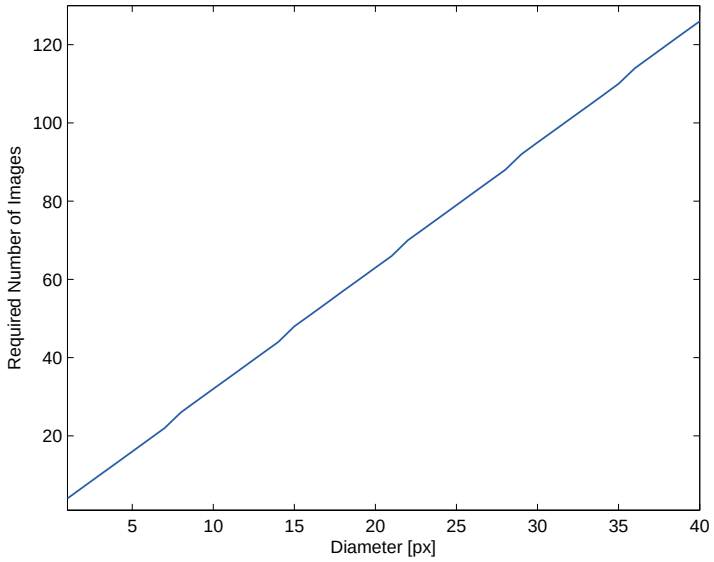
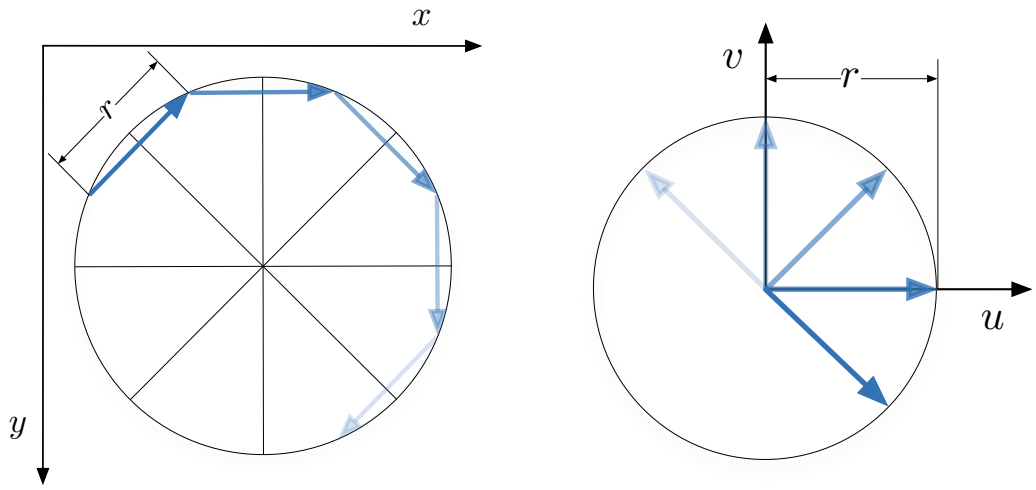


Figure 3.33: Required number of images to ensure subpixel motion between evenly spaced aperture positions plotted as a function of pixel diameter.

It should be mentioned, that the calculation of the integer pixel displacements can be avoided by using enough AWS aperture positions. Hence, the disparity between the images is reduced to being subpixel. Figure 3.33 relates a given maximum diameter to according numbers of evenly spaced aperture positions.
According to Frigerio, the required number of aperture positions can be significantly reduced, if the sampling intervals are not uniform. In order to do so, one has to place the second aperture position very close to the anchor position. By using these two aperture positions an accurate enough rotation diameter estimate can be calculated, which allows the other images to be sampled at greater angles.

3.6.8 Frigerio's Multi Image AWS Algorithm with long spatio-temporal Filter

Although it might be an incorrect assumption for general AWS applications, it is still worthwhile to consider first the situation how images undergoing uniform motion can be processed by an AWS system using long filters.
As a consequence of the uniform motion over the whole image the radius of the circular image motion will be uniform too. Also note, that due to the uniform sampling positions the motion is also periodic. This means that the image captured by the last aperture position will smoothly flow into the image captured by the first aperture position. This periodicity allows the use of long spatiotemporal derivative filters to calculate the N velocities. Note, because the motion is uniform across each image, the N velocities can be calculated at the same image position.
In order to calculate the radius R, Frigerio defined a spatiotemporal interrogation cuboid G



(a) Image space.

(b) Velocity space.

Figure 3.34: Illustration of the motion model used in Frigerio's multi image AWS approach with long spatio-temporal filter. Figure (a) shows the velocities between consecutive images in the image space. Here the motion describes the circumcircle of a blur spot. Figure (b) shows the same velocities in the velocity space, where they describe a parameterization of a circle with radius $r = R \cdot L$.

of size $8 \times 8 \times N$ pixels over which he solved the constant brightness equation for the rotation in the least squares sense. The minimization problem was formulated as follows:

$$\min_{R} \sum_{i,j,k} \left[G_x(i,j,k)\, u(k) + G_y(i,j,k)\, v(k) + G_t(i,j,k) \right]^2 \tag{3.70}$$

with

$$u(k) = L\, R\, sin\left(\alpha_k\right),$$
$$v(k) = -L\, R\, cos\left(\alpha_k\right),$$

where

$$L = \frac{2\pi}{N} \quad \text{and} \quad \alpha_k = \frac{2\pi}{N} k. \tag{3.71}$$

By calculating the derivative with respect to R and by setting it to zero, Frigerio obtained the following equation for the radius R:

$$R = \left[\sum_{i,j,k} G_x^2(i,j,k)\, sin^2(\alpha_k) - 2G_x(i,j,k)\, G_y(i,j,k)\, sin(\alpha_k)\, cos(\alpha_k) + G_y^2(i,j,k)\, cos^2(\alpha_k)\right]^{-1}$$
$$\frac{N}{2\pi}\left[\sum_{i,j,k} G_t(i,j,k)\,(G_y(i,j,k)\, cos(\alpha_k) - G_x(i,j,k)\, sin(\alpha_k))\right].$$

Note, that this equation assumes that R remains constant throughout the spatiotemporal interrogation cuboid, and therefore it is not valid in the case of non-uniform motion. By introducing non-uniform motion the actual radius of motion, R, at any spatial position can change in time. In order to handle non-uniform motion the spatiotemporal interrogation cuboid must be made to track the target of interest through time.
According to Frigerio, this algorithm creates a much noisier 3D surface model, than the multi image AWS approach from Section 3.6.7.

Chapter 4

Methodology

Contents

Based on the Frigerio multi image AWS approach with long spatio-temporal filter (cf Section 3.6.8), we present in this section a global version of this method, which we refer to as global AWS. The main motivation for introducing a global version is to reduce the noise effects of the models, calculated with the Frigerio multi image AWS approach with long spatio-temporal filter. Therefore, the global AWS approach assumes, that the blur-circle-radius varies smoothly almost everywhere in the image. Further, we assume that reflectance varies smoothly and the illumination is uniform across the surface.
For a better understanding, we first present a version of the global AWS approach, that uses the L^2-norm to weight the data-term. This version is referred to as L^2 global AWS (cf Section 4.1). In a second step, a robust version of the global AWS approach is presented, that is based on total variation (TV) in the regularization-term and the robust L^1-norm in the data-term. We present a numerical scheme to solve the according TV-L^1 global AWS problem (cf Section 4.2). This iterative method is based on a dual formulation of the TV energy and a point-wise thresholding step. Additional, the calculation is embedded into a coarse-to-fine warping approach, which avoids convergence to unfavorable minima. Moreover, the TV-L^1 global AWS approach is capable of parallel processing and hence the approach can be accelerated by graphics processing units (GPUs).
Experimental results for synthetic image sequences are then presented in Chapter 5.

4.1 L^2 Global AWS

In order to extend the Frigerio multi image approach with long spatio-temporal filter to a global version, we add a smoothness constraint. Therefore, neighboring points on the object have similar radii and the flow field varies smoothly almost everywhere. The according minimization problem is written as follows:

$$E = \int_{\Omega} \sum_{k} \left[I_x(x,y,k)\, u(k) + I_y(x,y,k)\, v(k) + I_t(x,y,k) \right]^2 + \lambda \|\nabla R\|^2 \, d\Omega \tag{4.1}$$

with

$$u(k) = L\, R\, sin\,(\alpha_k)\,,$$
$$v(k) = L\, R\, cos\,(\alpha_k)\,,$$

where

$$L = \sqrt{2 - 2cos\left(\frac{2\pi}{N}\right)} \quad \text{and} \quad \alpha_k = \frac{2\pi}{N} k.$$

λ represents a parameter, reflecting the influence of the smoothness term. Note that compared to Frigerio's approach we use a more accurate approximation of L.

Equation 4.1 can be minimized, using the calculus of variations:

$$\frac{\partial f}{\partial R} - div\left(\frac{\partial f}{\partial(\nabla R)}\right) = 0 \tag{4.2}$$

where f defines the integrand of E. In the following calculation we use $I(k)$ as the sloppy notation for $I(x,y,k)$. Thus,

$$\begin{aligned} \frac{\partial f}{\partial R} =& 2\, L^2\, R \sum_k \left[I_x^2(k)\, sin^2(\alpha_k) + 2\, I_x(k)\, I_y(k)\, sin(\alpha_k)\, cos(\alpha_k) + I_y^2(k)\, cos^2(\alpha_k) \right] + \\ & 2\, L\, I_t(k) \sum_k \left[I_x(k)\, sin(\alpha_k) + I_y(k)\, cos(\alpha_k) \right] \end{aligned} \tag{4.3}$$

and

$$div\left(\frac{\partial f}{\partial(\nabla R)}\right) = \lambda\, 2\, \nabla^2\, R. \tag{4.4}$$

Next, we calculate the second order derivative of R by using a Laplacian approximation. The approximation can be calculated by setting $\beta = 1/3$ in the following general Laplacian mask:

$$\begin{bmatrix} \frac{\beta}{1+\beta} & \frac{1-\beta}{1+\beta} & \frac{\beta}{1+\beta} \\ \frac{1-\beta}{1+\beta} & \frac{-4}{1+\beta} & \frac{1-\beta}{1+\beta} \\ \frac{\beta}{1+\beta} & \frac{1-\beta}{1+\beta} & \frac{\beta}{1+\beta} \end{bmatrix} \overset{\beta=1/3}{\Longrightarrow} 3 \begin{bmatrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & -1 & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{bmatrix}. \tag{4.5}$$

Therefore, the Laplacian approximation is calculated by subtracting the value at a certain point from a weighted average of according neighborhood points. Using this, we can approximate $\nabla^2 R$ as follows:

$$\nabla^2 R \approx 3(\bar{R} - R), \tag{4.6}$$

where $\bar{R}$ denotes the local neighborhood average. Substituting Equation 4.6 into Equation 4.4 leads to:

$$div\left(\frac{\partial f}{\partial(\nabla R)}\right) \approx \bar{\lambda}\, 2\, (\bar{R} - R), \tag{4.7}$$

where $\bar{\lambda} = 3\lambda$. In a final step, we substitute Equations 4.3 and Approximation 4.7 into Equation 4.2, and solve the resulting equation with respect to R:

$$\begin{aligned} R = &\left[L^2 \sum_k \left[I_x^2(k)\, sin^2(\alpha_k) + 2 I_x(k)\, I_y(k)\, sin(\alpha_k)\, cos(\alpha_k) + I_y^2(k)\, cos^2(\alpha_k) \right] + \bar{\lambda} \right]^{-1} \\ &\left[\bar{\lambda}\bar{R} - L \sum_k \left[I_x(k)\, sin(\alpha_k) + I_y(k)\, cos(\alpha_k) \right] I_t(k) \right]. \end{aligned} \tag{4.8}$$

As in the Frigerio approach, this problem is not valid for non-uniform motion. To handle non-uniform motion the derivative images I_x, I_y and I_t need to be warped according to the radius R.

4.2 TV-L^1 Global AWS

Due to the fact, that the quadratic estimator, used in Section 4.1, is expected to be noise sensitive, we also consider a more robust version of our global AWS approach by using a robust estimator for the data-term. Moreover, we also use the Euclidean norm in the smoothness-term, which leads to edge-preserved and smoothed depth-maps. Therefore, the minimization problem is now given by:

$$\min_R \int_\Omega \lambda \sum_k |I_x(k)\, u(k) + I_y(k)\, v(k) + I_t(k)| + \|\nabla R\|\, d\Omega, \tag{4.9}$$

where $u(k)$ and $v(k)$ are defined in Equation 4.1. Note, that the data-term is weighted by λ instead of the smoothness-term. In order to solve this minimization problem, we first

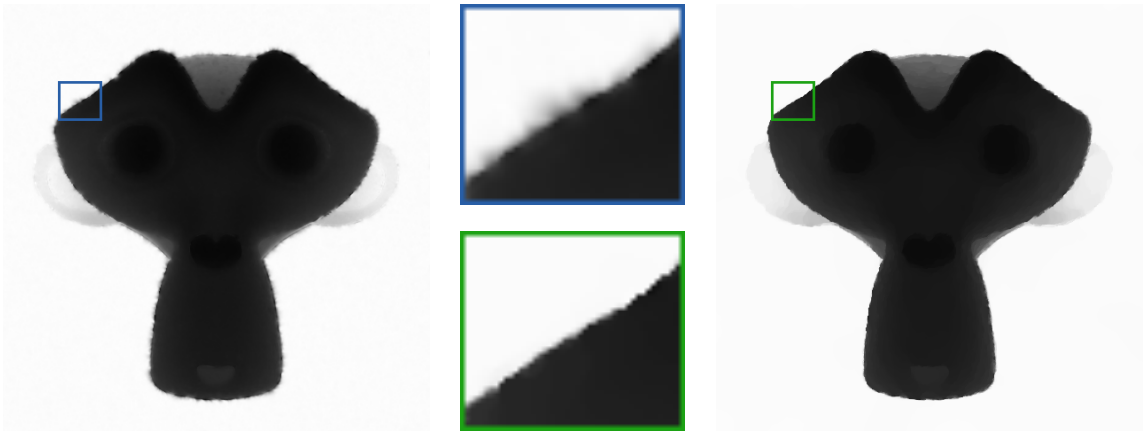


Figure 4.1: Example depth-maps of the Suzanne-Scene (compare Chapter 5). The left image shows a depth-map calculated with the L^2 global AWS approach from Section 4.1. The right image shows a depth map obtained by using the TV-L^1 global AWS approach. The closeup views clearly show that the right depth-map is edge-preserving, whereas the left depth-map is not.

introduce auxiliary variables $\bar{R}_k$ for $1 \leqslant k \leqslant N$, and we propose to minimize the following approximation of Equation 4.9:

$$\min_{R,\bar{R}_k} \int_\Omega \lambda \sum_k |I_x(k)\,\bar{u}(k) + I_y(k)\,\bar{v}(k) + I_t(k)| + \frac{1}{2\theta} \sum_k \left(R - \bar{R}_k\right)^2 + \|\nabla R\|\, d\Omega \tag{4.10}$$

with

$$\bar{u}(k) = L\,\bar{R}_k\, sin\,(\alpha_k)\,,$$
$$\bar{v}(k) = L\,\bar{R}_k\, cos\,(\alpha_k)\,,$$

where θ is a small constant, so that each $\bar{R}_k$ is a close approximation of R. The minimization is done by alternating update steps. First each $\bar{R}_k$ for $1 \leqslant k \leqslant N$, and second R is updated. Therefore, first one has to solve the following for $1 \leqslant k \leqslant N$ and for fixed R:

$$\min_{\bar{R}_k} \int_\Omega \lambda |\underbrace{I_x(k)\,\bar{u}(k) + I_y(k)\,\bar{v}(k) + I_t(k)}_{=:\rho(\bar{R}_k)}| + \frac{1}{2\theta} \left(R - \bar{R}_k\right)^2 d\Omega. \tag{4.11}$$

In a second step, one has to minimize

$$\min_{R} \int_\Omega \frac{1}{2\theta} \sum_k \left(R - \bar{R}_k\right)^2 + \|\nabla R\|\, d\Omega \tag{4.12}$$

for fixed $\bar{R}_k$, $1 \leqslant k \leqslant N$. The minimization problem in Equation 4.11 can be solved by a thresholding step, since it is a point-wise problem. Hence, the solution is given by the

following thresholding scheme:

$$\bar{R}_k = R + \begin{cases} \lambda\theta U_k & \text{if } \rho(R) < -\lambda\theta U_k^2 \\ -\lambda\theta U_k & \text{if } \rho(R) > \lambda\theta U_k^2 \\ -\rho(R)/U_k & \text{if } |\rho(R)| \leqslant \lambda\theta U_k^2, \end{cases} \tag{4.13}$$

where

$$U_k = I_x(k)\, L\, sin(\alpha_k) + I_y(k)\, L\, cos(\alpha_k). \tag{4.14}$$

This can be easily proved by analyzing the three different cases $\rho(\bar{R}_k) < 0$, $\rho(\bar{R}_k) > 0$, and $\rho(\bar{R}_k) = 0$.
The second problem (4.12) is very similar to the variation based image denoising model of Rudin et al. [27], and can therefore be solved in a similar way. Thus, we first introduce the dual variable p (cf Fig. 4.2), so that

$$\|\nabla R\| = \underset{\|p\| \leqslant 1}{max} \{\nabla R \cdot p\}. \tag{4.15}$$

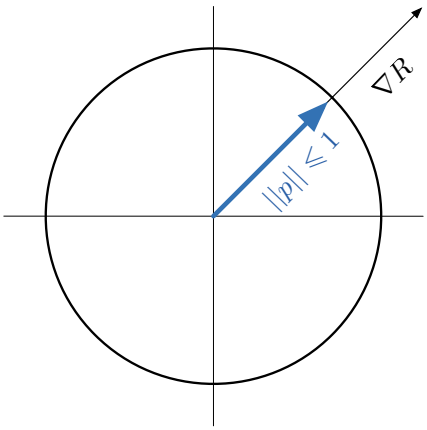


Figure 4.2: Illustration of the dual variable p used in Equation 4.15.

By substituting this into Equation 4.12 we obtain

$$\underset{R}{min}\ \underset{\|p\| \leqslant 1}{max} \int_\Omega \frac{1}{2\theta} \sum_k \left(R - \bar{R}_k\right)^2 + \nabla R \cdot p \, d\Omega. \tag{4.16}$$

Next, we use the divergence theorem

$$-\int_\Omega u(\nabla \cdot p) d\Omega = \int_\Omega p \cdot (\nabla u) d\Omega \tag{4.17}$$

to obtain a point-wise minimization problem in R:

$$\min_{R} \max_{\|p\| \leqslant 1} \int_{\Omega} \frac{1}{2\theta} \sum_{k} \left(R - \bar{R}_k\right)^2 - R(\nabla \cdot p)\, d\Omega. \tag{4.18}$$

This can be solved, for fixed p, by taking the derivative with respect to R and setting it to zero.

$$\frac{\partial}{\partial R} \left\{ \int_{\Omega} \frac{1}{2\theta} \sum_{k} \left(R - \bar{R}_k\right)^2 - R(\nabla \cdot p)\, d\Omega \right\} = \frac{1}{\theta} \sum_{k} \left(R - \bar{R}_k\right) - div(p) = 0, \tag{4.19}$$

which leads to the following result:

$$R = \frac{1}{N} \sum_{k} \bar{R}_k + \frac{\theta}{N} div(p). \tag{4.20}$$

In a second step, one has to update p for fixed R. This is done by a gradient ascent update scheme, where

$$\frac{\partial}{\partial p} \left\{ \int_{\Omega} \frac{1}{2\theta} \sum_{k} \left(R - \bar{R}_k\right)^2 + \nabla R \cdot p\, d\Omega \right\} = \nabla R. \tag{4.21}$$

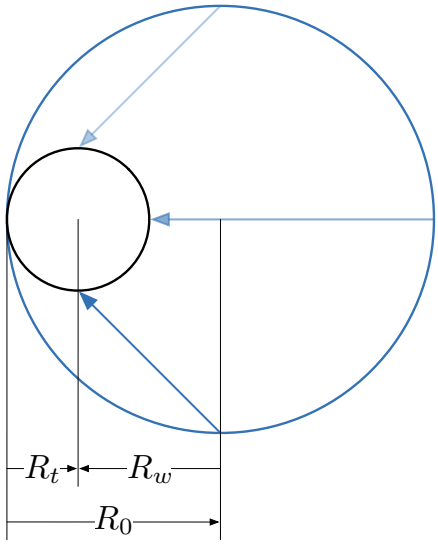
Finally, one can solve (4.12) using the following iteration, where n indicates the iteration step:

$$\begin{aligned} R^{n+1} &= \frac{1}{N} \sum_{k} \bar{R}_k^n + \frac{\theta}{N} div(p^n) \\ p^{n+1} &= \frac{p^n + \tau \nabla R^{n+1}}{max\{1, \|p^n + \tau \nabla R^{n+1}\|\}}, \end{aligned}$$

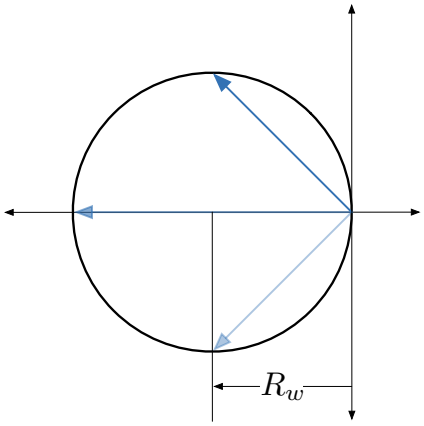
where τ is the step length.

4.3 Coarse-to-Fine Approach

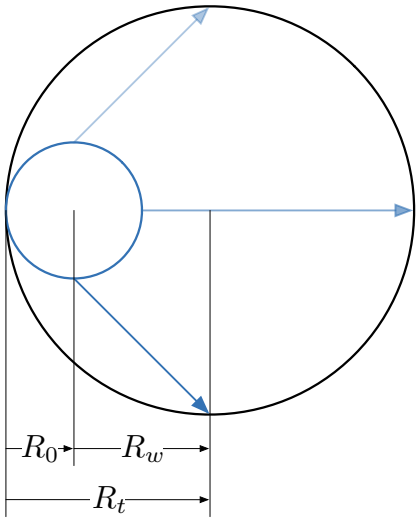
The minimization problems (4.9) and (4.1) are only valid for small displacements between consecutive aperture positions. Thus, the minimization is embedded into a coarse-to-fine warping approach. This avoids convergence to unfavorable local minima. Therefore an image-pyramide with a downsampling factor of two is used (see Figure 4.4). In each level of the image-pyramide the images are warped according to a given radius R_0. Hence, only the remaining radius $R - R_0$ needs to be calculated. This warping steps are valid by simply considering Figure 4.3. It illustrates the two possible situations, where the current blur-circle is shrunk (cf Fig. 4.3(a)) or extended (cf Fig. 4.3(c)). In both figures the blue circle with radius R_0 depicts the current blur-circle and the black circle with radius R_t defines a better approximation of the true blur-circle. By warping the current blur-circle to the anchor



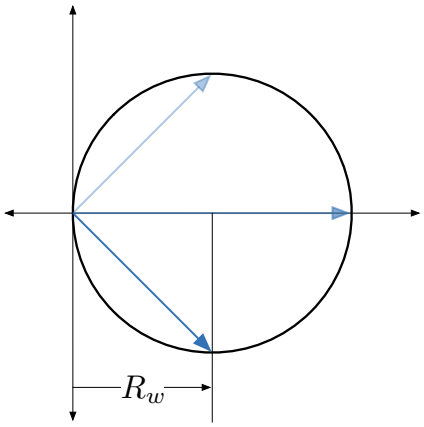
(a) Shrinking situation.



(b) Shrinking situation after warping.

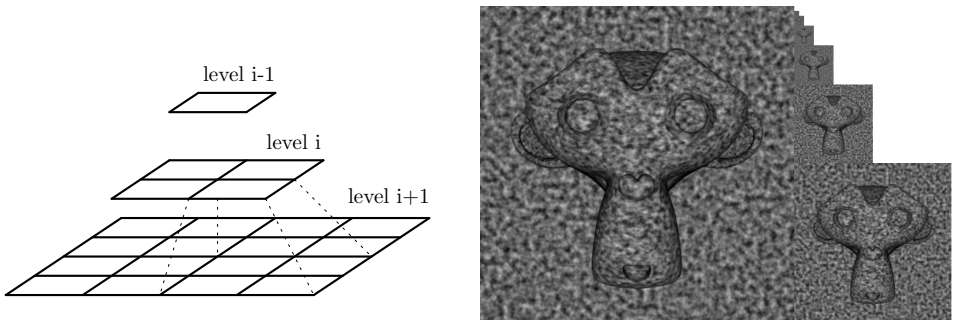


(c) Extending situation.



(d) Extending situation after warping.

Figure 4.3: Illustration of the coarse-to-fine warping strategy. Figures (a) and (c) show the situations of shrinking and extending the current blur-circle, respectively. In both situations the blue circle with radius R_0 depicts the current blur-circle and the black circle with radius R_t defines a better approximation of the true blur-circle. Figures (b) and (d) show the situation after warping the current blur-circle to the anchor position (9 o'clock position). Here, the new calculated radius R_w has a negative orientation in Figure (b) and a positive orientation in Figure (d). Therefore, in both situations R_t can be calculated by the sum of $R_0 + R_w$.



(a) Architecture. (b) Example.

Figure 4.4: Illustration of an image pyramid [2]. Figure (a) shows the architecture of an image pyramid with a downsampling factor of two. Figure (b) shows an example for an image pyramid with seven levels.

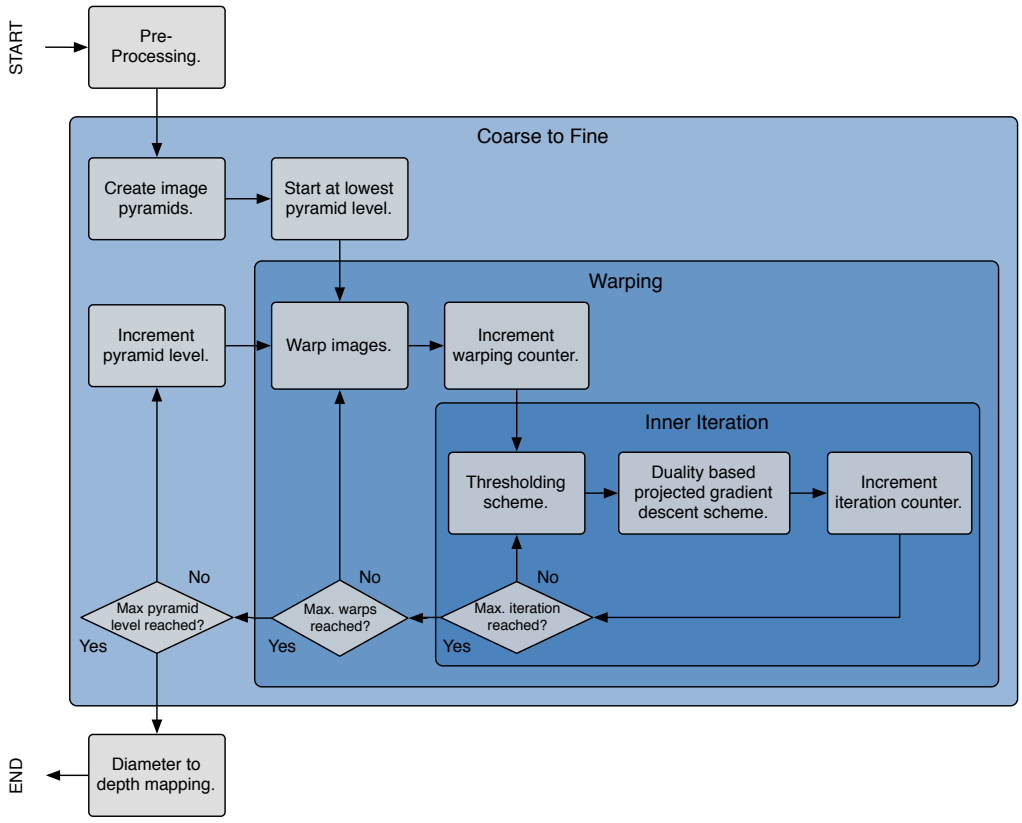


Figure 4.5: Flowchart of the proposed TV-L^1 global AWS algorithm.

position (9 o'clock position) one obtains the situations shown in Figure 4.3(b) and 4.3(d), respectively. Here, the new calculated radius R_w has a negative orientation in Figure 4.3(b) and a positive orientation in Figure 4.3(d). Thus, in both situations R_t can be calculated by the sum of $R_0 + R_w$.
As a result we can sum up the whole TV-L^1 global AWS algorithm by the flowchart shown in Figure 4.5. We start with some pre-processing, e.g. to reduce image noise. Next, we create the image pyramids, which are then processed from top to bottom. At each level the images are warped according to the coarse solution. The TV-L^1 global AWS problem is then solved in an inner iteration. After processing all levels one obtains the diameter result for each pixel. In a last step, these diameter results are used to calculate the actual depth values.

4.4 Acceleration by Graphics Processing Units

Due to the fact, that the methods presented in Section 4.1 and 4.2 work on regular grids, they can be effectively accelerated by Graphics Processing Units (GPUs). A GPU is typically only used for computer graphics computation. The approach of using a GPU to perform general purpose computing is referred to as GPGPU, GPGP or GP^2 (General Purpose Computing on Graphics Processing Units).

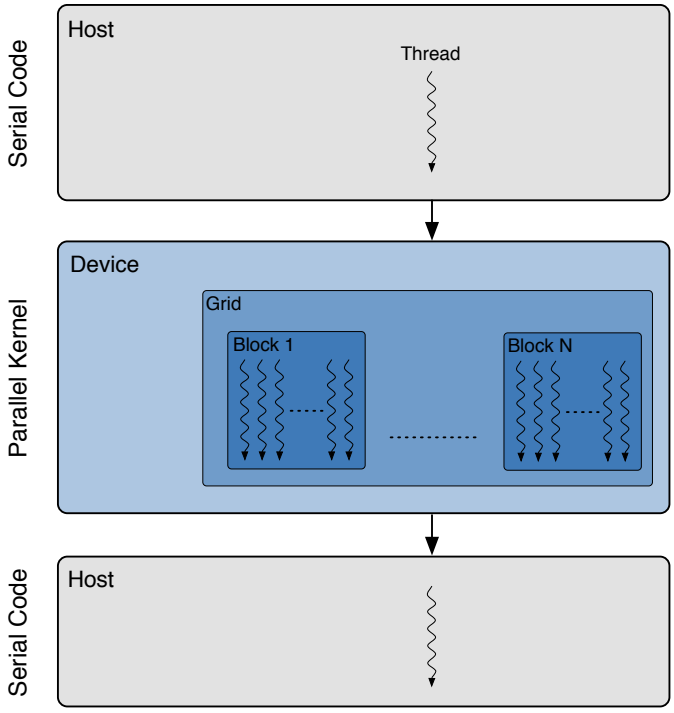


Figure 4.6: Illustration of the heterogeneous programming in CUDA. CUDA enables one to use parallel kernels within the serial code. These parallel kernels execute the code in many device threads across multiple processing elements on the GPU, which can provide large speedups.

By using e.g. CUDA[1] (Compute Unified Device Architecture) one gets access to the computing engine in NVIDIA GPUs. CUDA is a scalable parallel programming model and a software environment for parallel computing developed by NVIDIA. It enables one to create serial programs with parallel kernels (cf Fig. 4.6). The serial code is executed in a host thread on the CPU. The parallel kernel code is executed in many device threads across multiple processing elements on the GPU.
By using the parallel processing capabilities of GPUs one can achieve a large performance benefit for applications, suitable for parallel processing. We are not providing an exact performance analysis for a CUDA implementation of our TV-L^1 global AWS approach. In our implementation only the inner iteration (cf Fig. 4.5) is implemented using CUDA on the GPU. Compared to a CPU implementation the CUDA implementation of this part of the algorithm brought a speedup S of about 50 for images of size 512×512 pixels, where speedup S is given by

$$S = \frac{t_{CPU}}{t_{GPU}}. \tag{4.22}$$

t_{CPU} is the execution time of the sequential implementation on the CPU, and t_{GPU} is the execution time of the parallel implementation on the GPU.

4.5 Conclusion

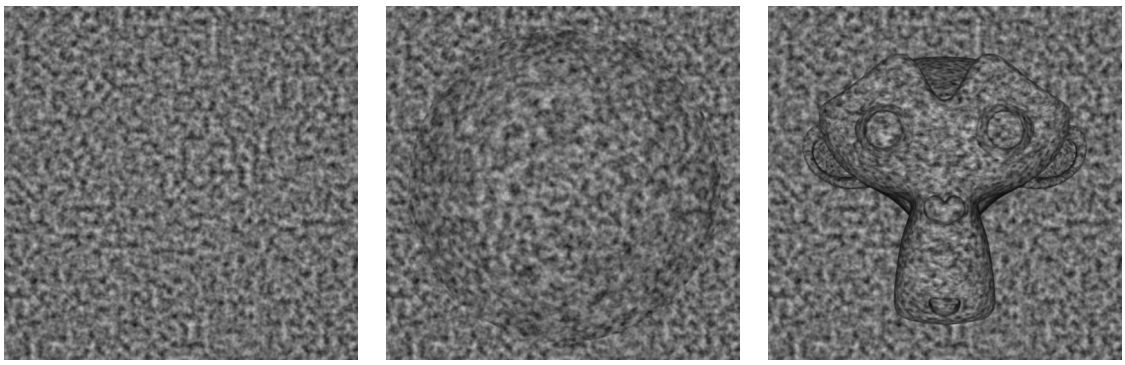
In this chapter we presented a global approach to calculate the image-rotation diameter generated by a rotating AWS mask. First, in Section 4.1 we described a solution, which makes use of a quadratic estimator. We showed that this version is very easy to minimize using the calculus of variations. However, due to the quadratic estimator, this version is expected to be noise sensitive. Therefore, we provided a robust version using a TV-L^1 energy functional in Section 4.2. We showed, how the minimization is done by using a numerical scheme. Furthermore, we also solved the problem of warping the derivative images according to the radius R by using a coarse-to-fine strategy described in Section 4.3. Finally, we mentioned, that the TV-L^1 global AWS approach has potential for parallel processing. Therefore it is recommended to use GPGPU approaches to speedup the computation.

[1] http://developer.nvidia.com/CUDA

Chapter 5

Experiments

Contents



(a) Plane-Scene. (b) Sphere-Scene. (c) Suzanne-Scene.

Figure 5.1: The images show the anchor images of three different scenes, used within conducted experiments.

In this chapter we evaluate the different Active Wavefront Sampling (AWS) approaches. First we start with the methods presented by Frigerio (cf Section 3.6.7 and 3.6.8) and then we evaluate the TV-L^1 global AWS approach (cf Section 4.2). Moreover, it should be mentioned, that we refer methods based on the Frigerio approach from Section 3.6.7 as local AWS.
The goal of this chapter is the comparison of the accuracy of the 3D reconstruction results acquired by the different AWS approaches. Therefore we use rendered image sequences of three different scenes, which we refer to as Plane-, Sphere-, and Suzanne-Scene. Figure 5.1

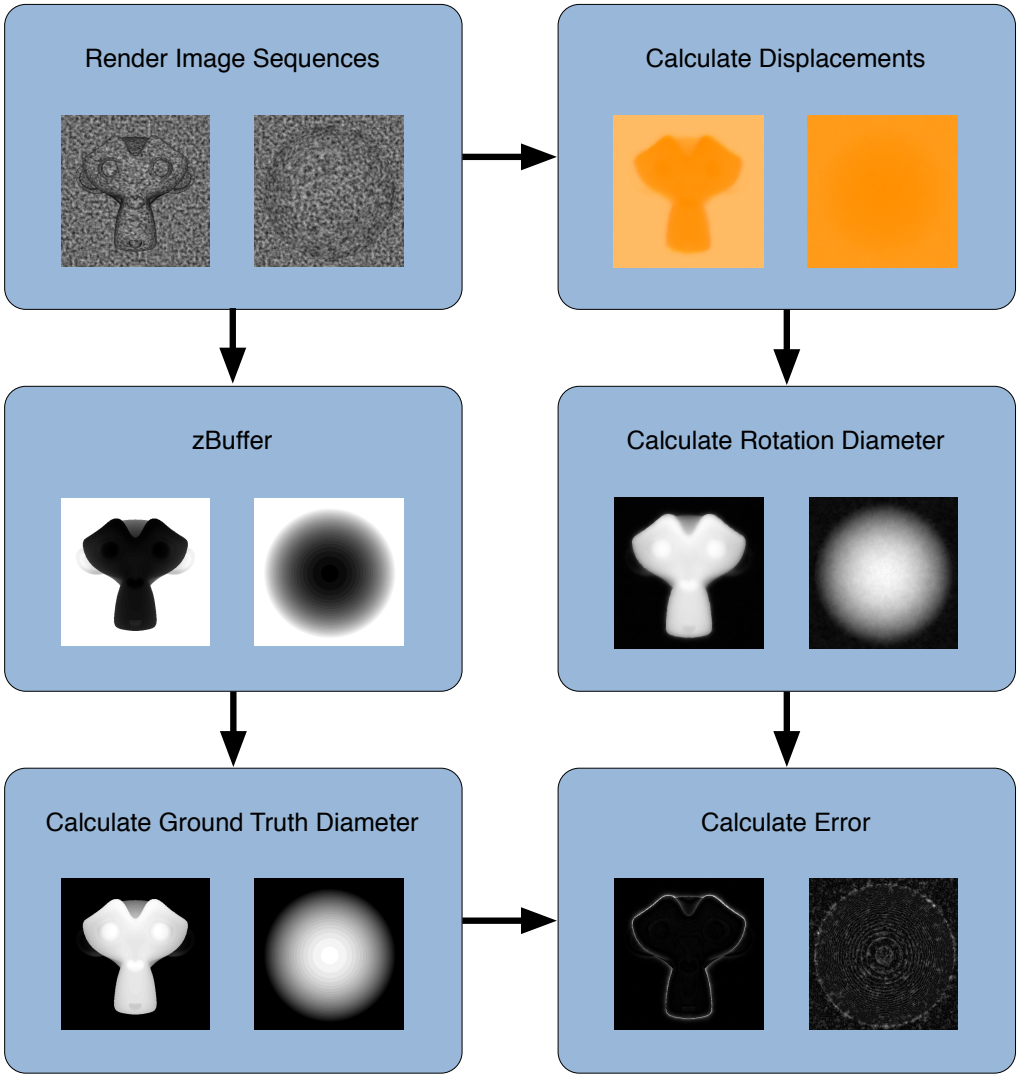


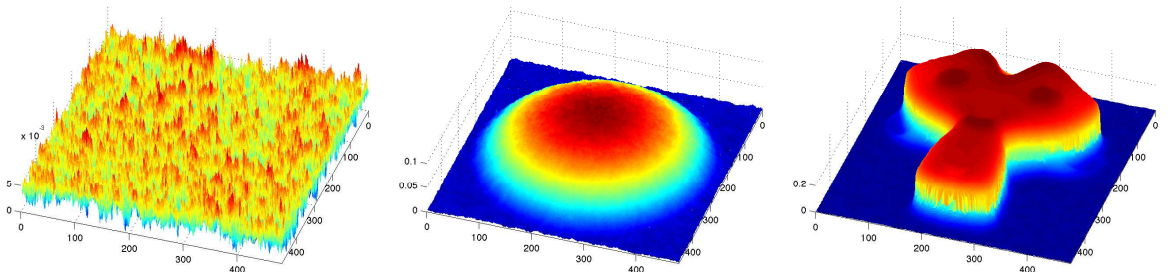
Figure 5.2: Illustration of the experimental procedure used to analyze the accuracy of the local AWS approach.

shows an example image for each scene. Each image sequence consists of 32 images of size 512×512 pixel, acquired by moving the camera position over evenly spaced positions on a circle. The different camera positions simulate the different aperture positions of an AWS system. Such a simulation of an AWS system is valid because of the comparison to stereo imaging described in Section 3.6.5. Moreover, all scenes use noise textures to improve the results and to allow the comparison between methods based on local and global optical flow techniques.

The Plane-Scene consists of a simple plane, that is placed parallel to the image plane. Thus the according ground truth rotation diameters are uniform over the whole image. In the Sphere-Scene the rotation diameters are no longer uniform. Here the rotation diameters are

high at the center of the image and decrease towards the margin of the image. Finally, the Suzanne-Scene presents not only non-uniform rotation diameter but also some occlusion effects.
The ground truth values for the depth are obtained by Blender's zBuffer values. These values are used to calculate ground truth diameters, which are then compared to the calculated results from the different AWS methods (cf Fig. 5.2).

5.1 Local AWS

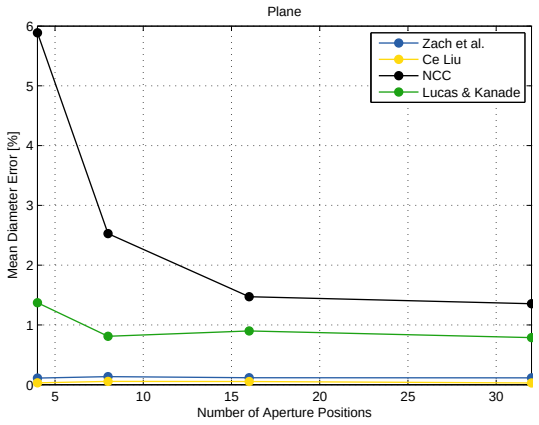


(a) Plane-Scene. (b) Sphere-Scene. (c) Suzanne-Scene.

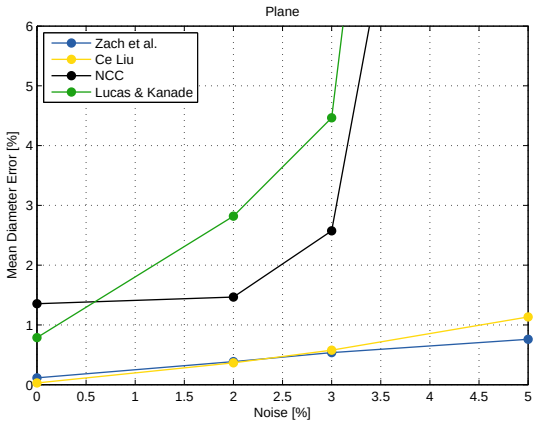
Figure 5.3: Depth reconstruction examples for (a) Plane-, (b) Sphere-, and (c) Suzanne-Scene, calculated with Frigerio's local AWS approach using a global optical flow presented by Zach et al. [32].

In order to obtain reference values for the accuracy, we first implement the local AWS approach, described in Section 3.6.7. The experimental procedure is illustrated in Figure 5.2. First, we render the image sequences for the Plane-, Sphere-, and Suzanne-Scene, using Blender[1]. Next, we calculate the displacements between the anchor image (first image) and the other images for each image sequence. The displacements are calculated using four different approaches. First, we use a blockmatching approach (NCC) to calculate the pixel displacements. Second, we use a local optical flow approach, presented by Lucas and Kanade [21]. Finally, we use two global optical flow approaches presented by Zach et al. [32] and Ce Liu [20]. The approach presented by Zach et al. is based on total variation (TV) and the Ce Liu approach is referred to as iterative re-weighted least squares (IRLS), because it iterates between computing the weight for non-linear terms and solving a least squares problem. After the displacement calculation, the diameter for each pixel in the anchor image is calculated using the Frigerio circle fitting method. We finally compare the calculated diameters with ground truth diameters, which are obtained by using Blender's zBuffer values. The mean relative diameter errors for the three different scenes are used for quantitative evaluation.
Due to the fact that we are mainly using differential based optical flow techniques to calculate the displacements between the images, we first assess the sensitivity to noise. Therefore, we

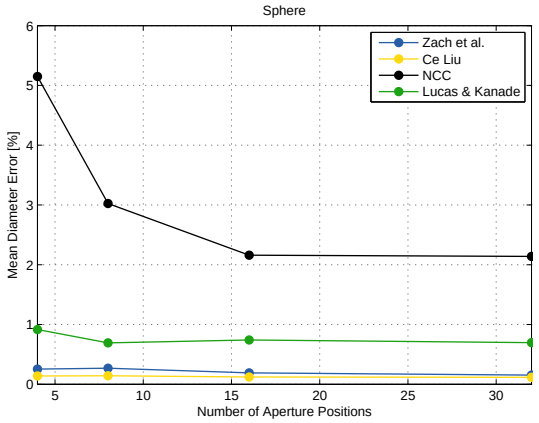
[1]Blender is a 3D graphics application released as free software under the GNU General Public License.



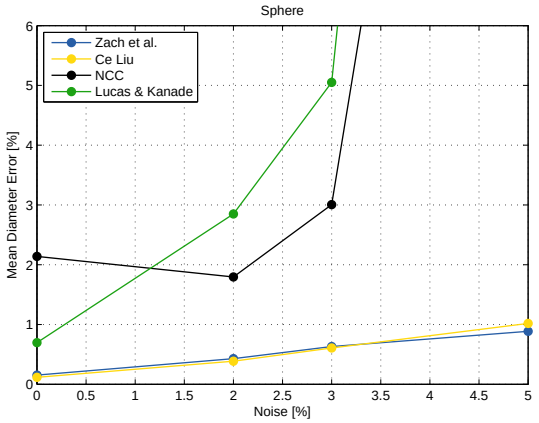
(a) Plane-Scene with 0% noise.



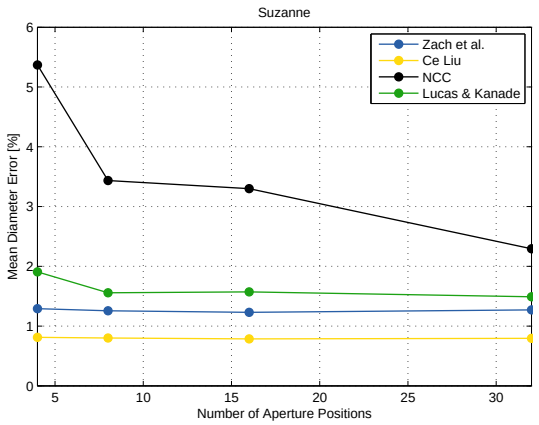
(b) Plane-Scene with 32 aperture positions.



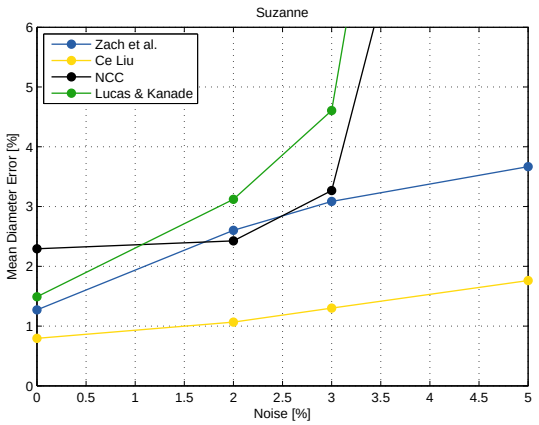
(c) Sphere-Scene with 0% noise.



(d) Sphere-Scene with 32 aperture positions.

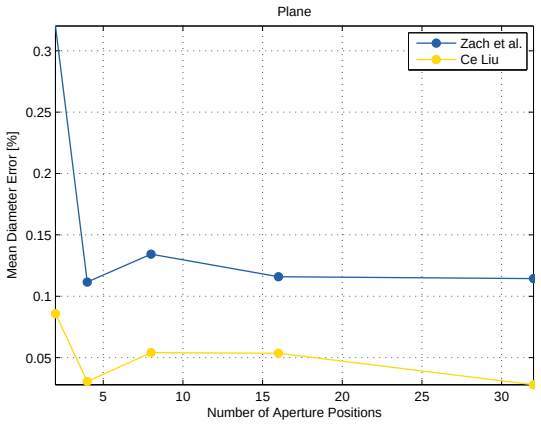


(e) Suzanne-Scene with 0% noise.

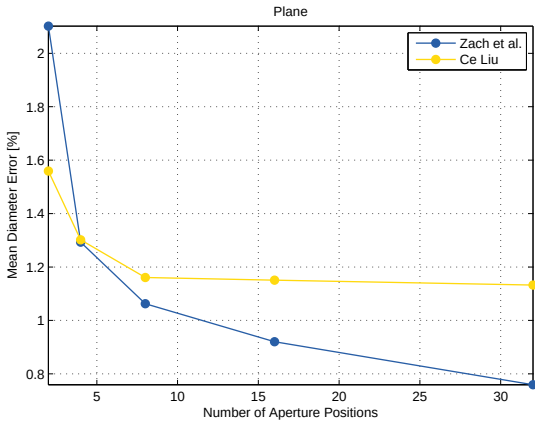


(f) Suzanne-Scene with 32 aperture positions.

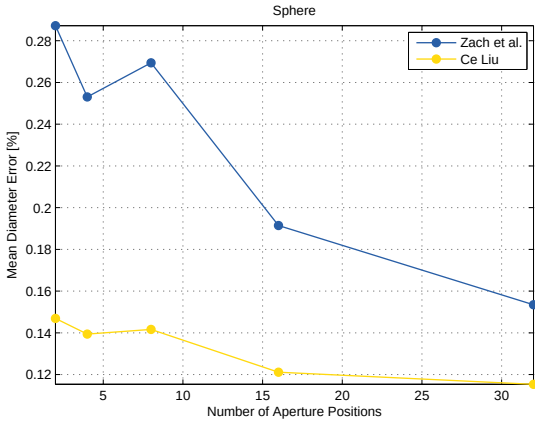
Figure 5.4: Comparison of four different optical flow techniques (Zach et al. [32], Ce Liu [20], NCC, Lucas and Kanade [21]) used within Frigerio's local AWS approach.



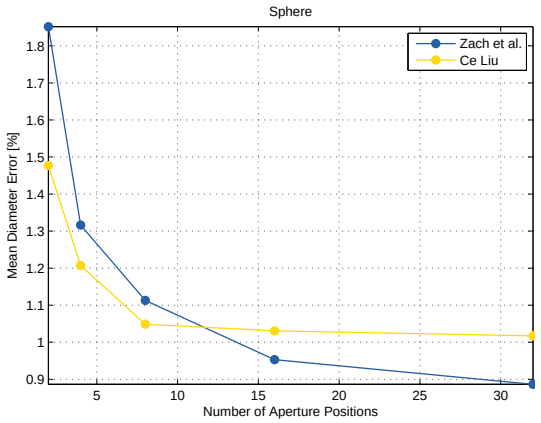
(a) Plane-Scene with 0% noise.



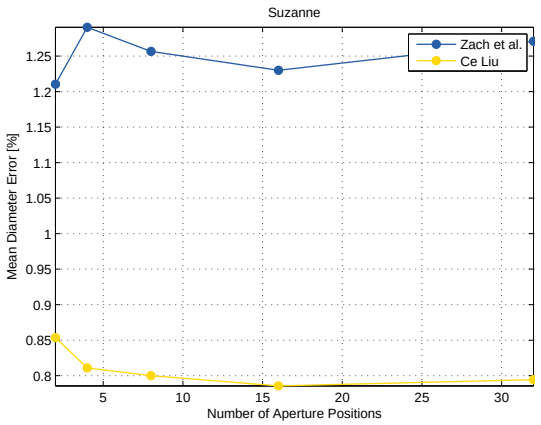
(b) Plane-Scene with 5% noise.



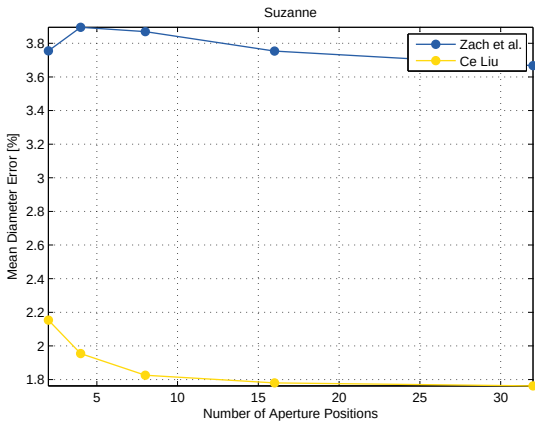
(c) Sphere-Scene with 0% noise.



(d) Sphere-Scene with 5% noise.



(e) Suzanne-Scene with 0% noise.



(f) Suzanne-Scene with 5% noise.

Figure 5.5: Comparison of two robust optical flow techniques (Zach et al. [32], Ce Liu [20]) used within the local AWS approach.

contaminate the rendered image sequences with additive Gaussian noise with a standard deviation of 0% to 5% of the image dynamic range. We further calculate the mean relative diameter error for the different noise levels and for different numbers of evenly spaced aperture positions. The results for all four optical flow techniques are shown in Figure 5.4.
As expected, the accuracy increases with increasing number of aperture positions. Accuracy decreases when noise increases. Because NCC and the Lucas-Kanade version provide poor accuracy results in the presence of image noise (see Figure 5.4), we concentrate on the two remaining techniques presented by Zach et al. and Ce Liu. In Figure 5.5 the accuracy results for the robust methods are shown for noise levels of 0% and 5%. Figure 5.5 shows increasing accuracy for an increasing number of aperture positions. Although this effect can be observed in the accuracy results of the original image sequences, it is more dominant in the presence of noise. This is shown in the right column of Figure 5.5. In the Suzanne-Scene this effect is not that obvious, because of partial occlusion, that is present in this scene.

5.2 Global AWS

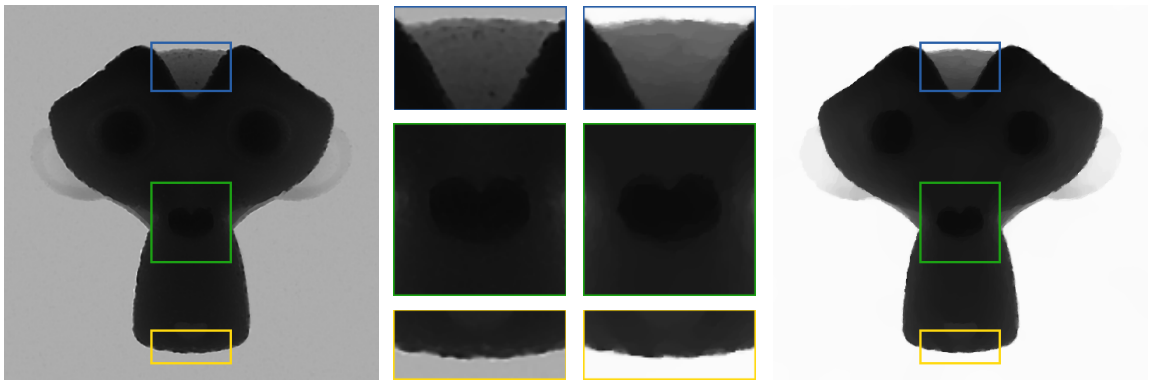


Figure 5.6: Depth reconstruction examples for the Suzanne-Scene. The left image shows the result for the Frigerio AWS approach (Section 3.6.8), where a 5×5 interrogation area was used. The right image shows the result for the TV-L^1 global AWS approach (Section 4.2), with $\lambda = 25$. The blue-closeup view depicts the presence of outliers in the Frigerio AWS approach, the green-closeup view shows a more homogenous area, and the yellow-closeup view shows a edge-region.

In this section we review the TV-L^1 global AWS approach presented in Section 4.2. We compare it, amongst others, to Frigerio's multi image AWS approach with long spatio-temporal filter (cf Section 3.6.8). As in the previous section, we use the Plane, Sphere, and Suzanne-Scene to test the approach.
The TV-L^1 global AWS approach as well as Frigerio's AWS algorithm are incorporated into a coarse-to-fine warping approach, described in Section 4.3. Thus it is sufficient to use two point derivative and interpolating filter. It is worth mentioning, that we slightly modifiy Frigerio's AWS approach, which means, that we use the same approximation

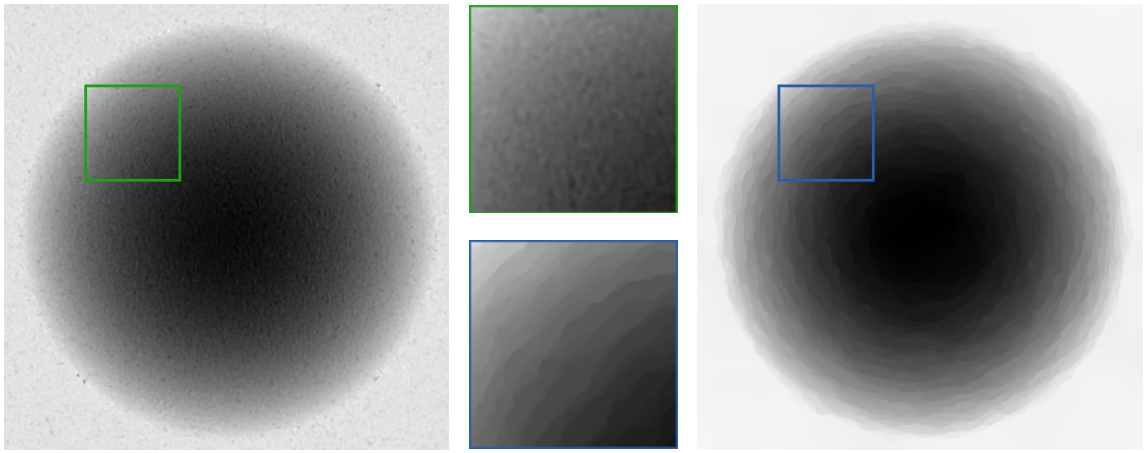
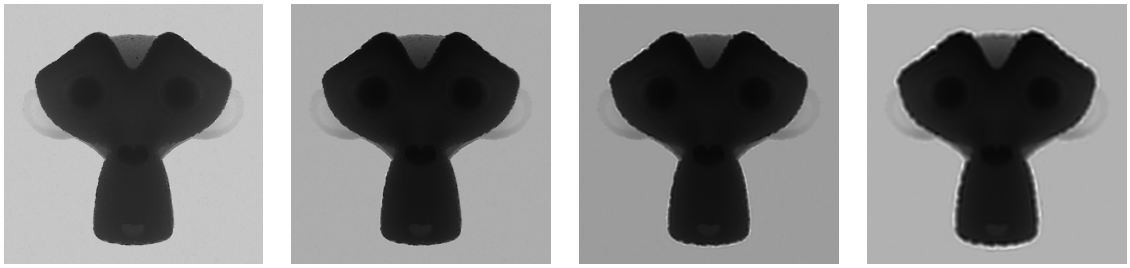


Figure 5.7: Depth reconstruction examples for the Sphere-Scene. The left image shows the result for the Frigerio AWS approach with long spatio-temporal filter (cf Section 3.6.8). The right image shows the result for the TV-L^1 global AWS approach (Section 4.2). The closeup views show, noise effects presented in the Frigerio AWS algorithm and the staircase effect in the TV-L^1 AWS approach.

$$L = \sqrt{2 - 2cos\left(\frac{2\pi}{N}\right)}$$

as in the TV-L^1 global AWS approach. Moreover, we also use a Gaussian weighting mask in order to weight the constraints in the center of the current neighborhood more highly, which reduces the visibility of quadratic patterns in the calculated depth map. However, by comparing the two approaches based on the Suzanne-Scene (cf Fig. 5.6), it can be seen, that Frigerio's algorithm produces outliers in the forehead area, whereas the global AWS algorithm produces a smoothed surface. Furthermore, by comparing the edge regions, one recognizes that Frigerio's algorithm produces a undesired spreading effect, whereas the global AWS algorithm seems to preserve the edges. Here the spread of the edges depends on the size of the local interrogation area (cf Fig. 5.8). Further, Frigerio's AWS algorithm seems to produce a much noisier surface than the global AWS algorithm. This can be seen by considering Figure 5.7. The noise effects can be reduced by increasing the local interrogation area (cf Fig. 5.8), but this will simultaneously increase the edge-spreading-effect. On the other hand, the TV-L^1 global AWS algorithm produces a staircase-effect (cf Fig. 5.7), because of the L^1 norm used within the data-term.

After this qualitative analysis we will now present a more quantitative analysis. Therefore, we first analyze the effect of the interrogation area in the Frigerio approach. Figure 5.10 shows the mean relative diameter errors for the three different scenes and for different numbers of aperture positions as a function of the interrogation area-size. It can be seen, that increasing the interrogation area increases the accuracy in the Plane and Sphere-Scene, but not in the Suzanne-Scene. This is due to the above mentioned edge-spreading-effect. Moreover, Figure

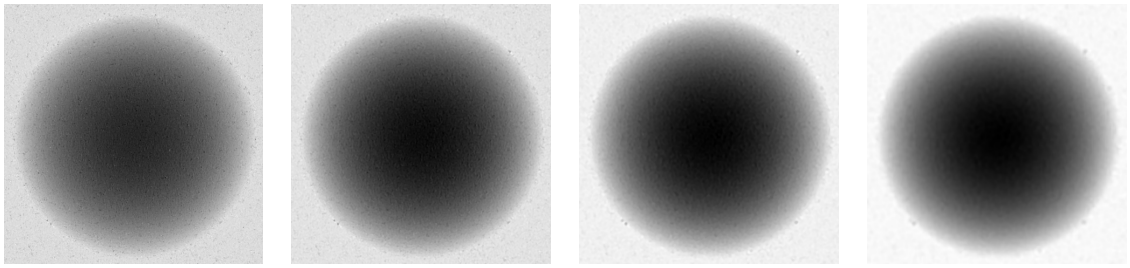


(a) Suzanne-Scene, 3×3 interrogation area.

(b) Suzanne-Scene, 5×5 interrogation area.

(c) Suzanne-Scene, 9×9 interrogation area.

(d) Suzanne-Scene, 15×15 interrogation area.



(e) Sphere-Scene, 3×3 interrogation area.

(f) Sphere-Scene, 5×5 interrogation area.

(g) Sphere-Scene, 9×9 interrogation area.

(h) Sphere-Scene, 15×15 interrogation area.

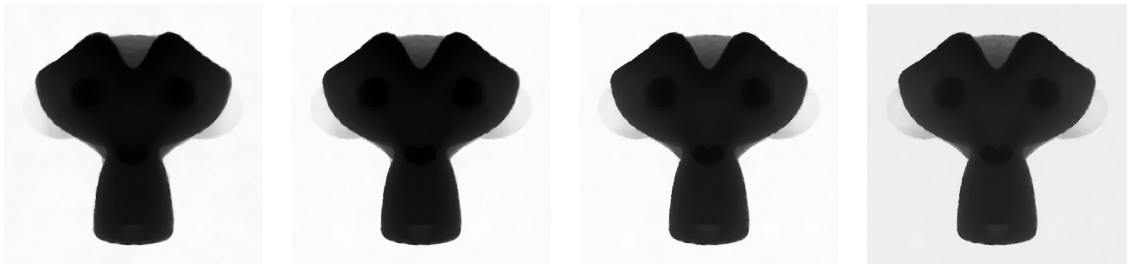
Figure 5.8: Depth reconstruction results for the Suzanne-Scene (first row) and Sphere-Scene (second row). The depth results are calculated using the Frigerio AWS approach (cf Section 3.6.8) with different interrogation area settings. It can be seen, that by increasing the interrogation area noise effects are reduced, but simultaneously the edge-spreading-effect is increased.

5.10 shows that increasing the number of aperture positions has a positive effect on the accuracy.

Next, we analyze the effect of the data-term weighting within the TV-L^1 global AWS approach. Figure 5.9 shows depth map results acquired by different λ settings. By increasing λ the data-term obviously gets more influence, therefore, finer structures become visible.

Figure 5.11 shows the mean relative diameter errors for the three different scenes and different numbers of aperture positions as a function of λ. As can be seen, increasing the number of aperture positions increases the accuracy. Moreover, by considering Figure 5.12 one recognizes that each aperture setting has a different optimal λ setting. This can be explained by the simple fact, that by increasing the number of aperture positions one simultaneously increases the terms of the sum within the data-term, which changes the weighting between data-term and smoothness-term.

The mean relative diameter errors for the three scenes are presented in Table 5.1. Here we

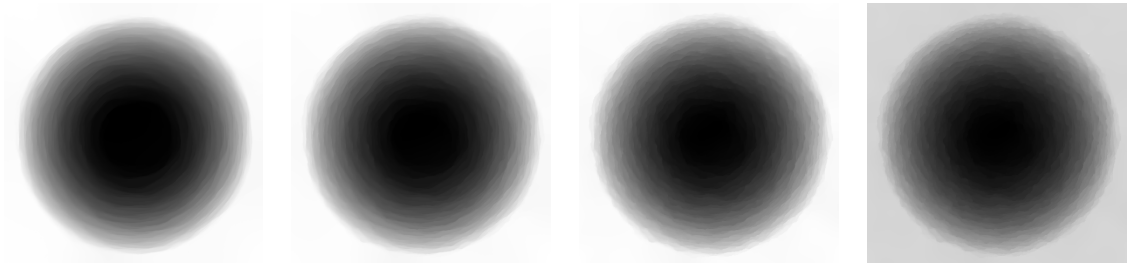


(a) Suzanne-Scene, $\lambda = 5$.

(b) Suzanne-Scene, $\lambda = 10$.

(c) Suzanne-Scene, $\lambda = 20$.

(d) Suzanne-Scene, $\lambda = 30$.



(e) Sphere-Scene, $\lambda = 5$.

(f) Sphere-Scene, $\lambda = 10$.

(g) Sphere-Scene, $\lambda = 20$.

(h) Sphere-Scene, $\lambda = 30$.

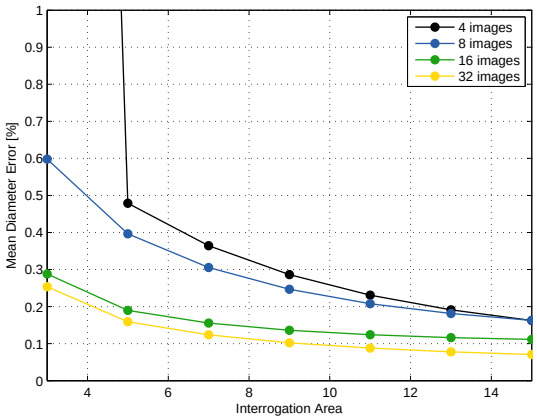
Figure 5.9: Depth reconstruction results for the Suzanne-Scene (first row) and Sphere-Scene (second row). The depth results are calculated using the TV-L^1 global AWS approach with different λ settings. It can be seen, that by increasing the influence of the data-term, finer structures become visible.

	Plane-Scene	Sphere-Scene	Suzanne-Scene
Local AWS (NCC)	1.3550	2.1386	2.2935
Local AWS (Lucas & Kanade)	0.7881	0.6956	1.4902
Local AWS (Zach et al.)	0.1144	0.1535	1.2708
Local AWS (Ce Liu)	0.0279	0.1154	0.7943
Frigerio AWS (with long filter)	0.0707	0.1216	1.0193
TV-L^1Global AWS	0.0205	0.1339	0.7886

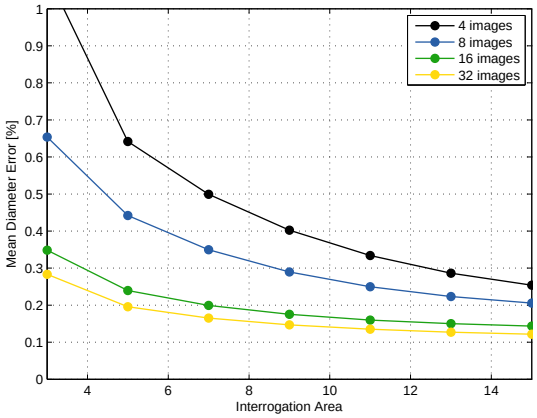
Table 5.1: Numerical results for the mean relative diameter errors for local and global AWS approaches.

compare the best results from Frigerio's AWS approach and the TV-L^1 global AWS approach to the local AWS approaches. It can be seen, that the TV-L^1 global AWS approach reaches slightly better results than Frigerio's AWS approach. Moreover, compared to the local AWS approaches, the TV-L^1 global AWS approach produces only marginally different errors.

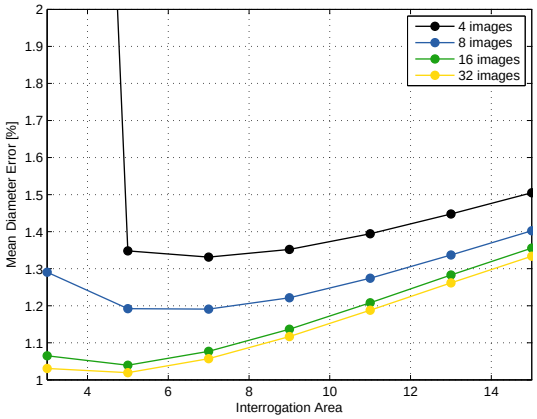
By considering additive image noise, we observe, that the TV-L^1 global AWS algorithm keeps the error within a reasonable limit, as long as the weighting between data-term and



(a) Plane-Scene.

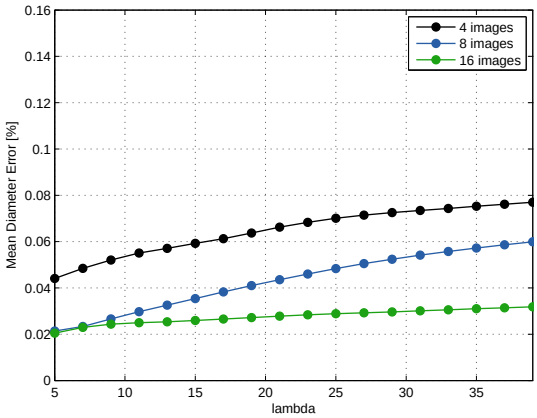


(b) Sphere-Scene.

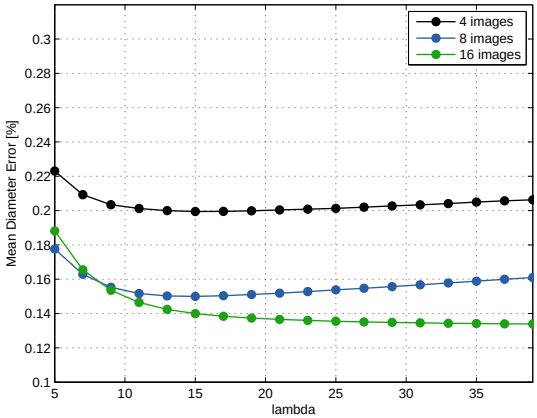


(c) Suzanne-Scene.

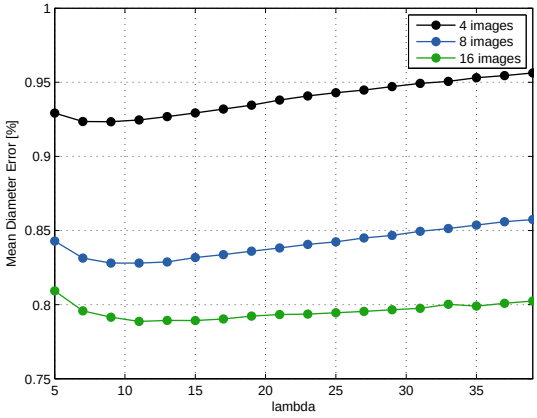
Figure 5.10: Accuracy analysis of the Frigerio multi image AWS approach with long spatio-temporal filter.



(a) Plane-Scene.

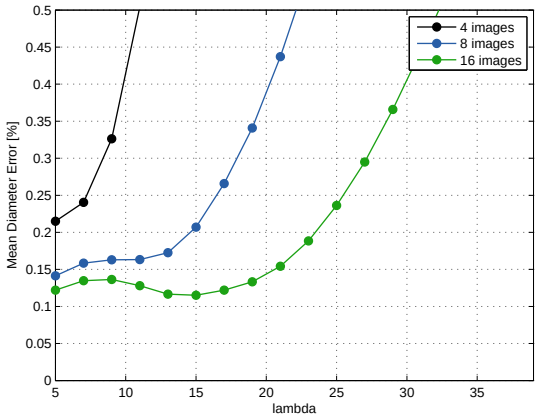


(b) Sphere-Scene.

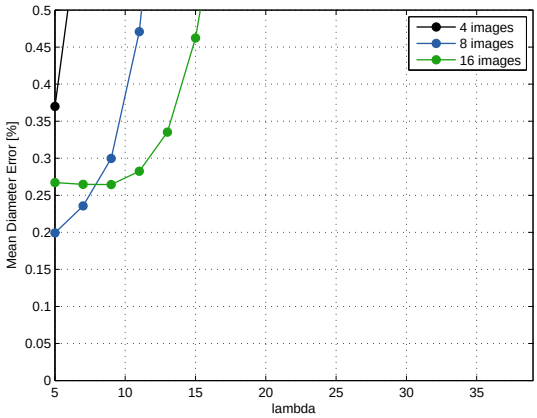


(c) Suzanne-Scene.

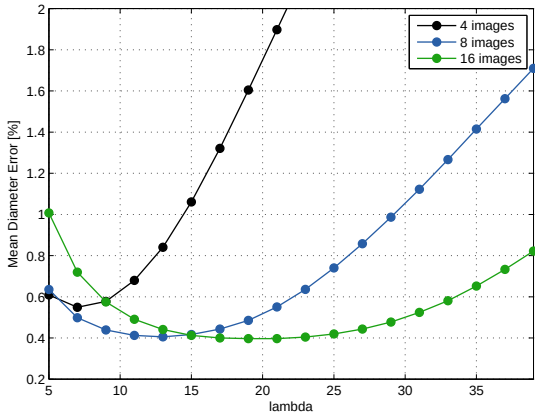
Figure 5.11: Accuracy analysis of the TV-L^1 global AWS approach.



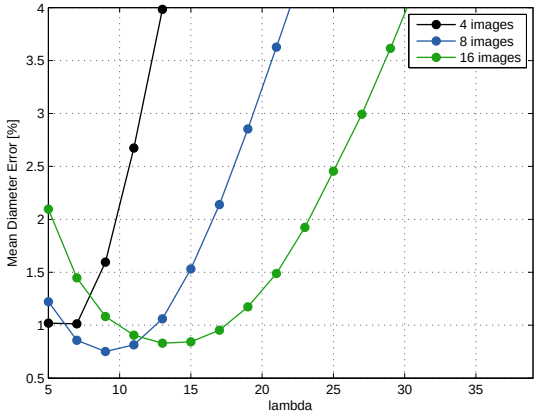
(a) Plane-Scene, 2% Noise.



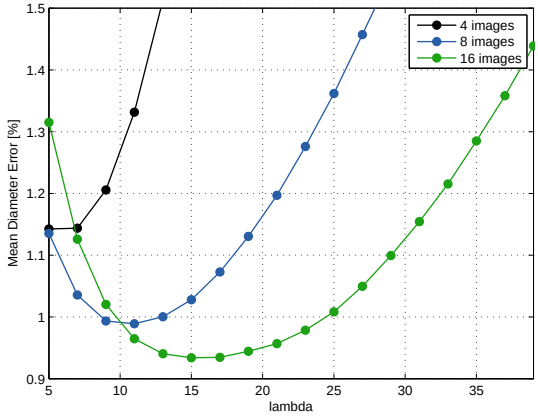
(b) Plane-Scene, 5% Noise.



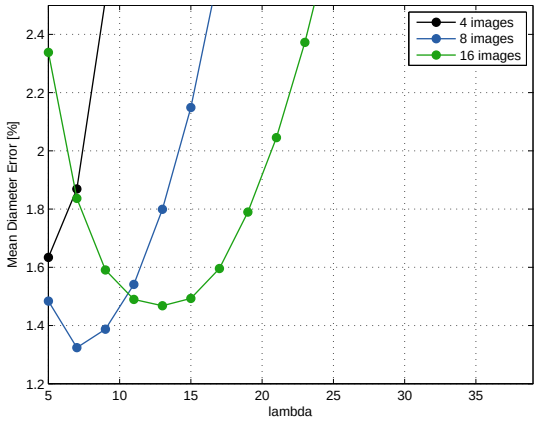
(c) Sphere-Scene, 2% Noise.



(d) Sphere-Scene, 5% Noise.

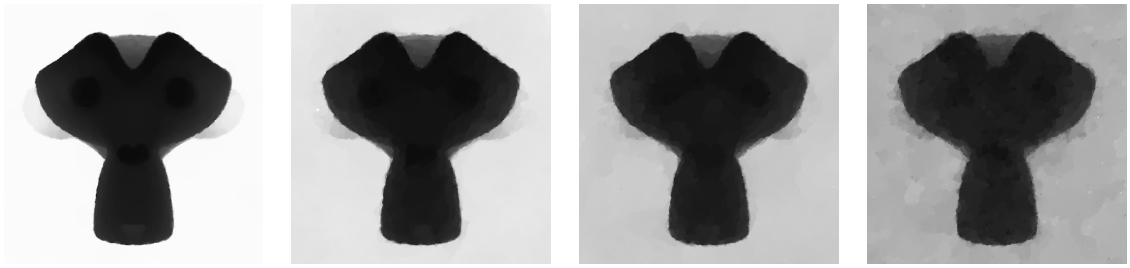


(e) Suzanne-Scene, 2% Noise.



(f) Suzanne-Scene, 5% Noise.

Figure 5.12: Accuracy analysis of the TV-L^1 global AWS approach.



(a) 0% Noise. (b) 2% Noise. (c) 3% Noise. (d) 5% Noise.

Figure 5.13: Depth-map examples for the Suzanne-Scene for different noise levels. Here the 32 images of the Suzanne-Scene are contaminated with additive Gaussian noise with a standard deviation of 0% to 5% of the image dynamic range, before the according depth-maps are calculated using the TV-L^1 global AWS approach.

	Plane-Scene	Sphere-Scene	Suzanne-Scene
Local AWS (Zach et al.)	0.3860	0.4288	2.6026
Local AWS (Ce Liu)	0.3651	0.3864	1.0661
TV-L^1Global AWS	0.1166	0.4413	0.9404

Table 5.2: Numerical results for the mean relative diameter errors for local and global AWS approaches with 2% image noise.

	Plane-Scene	Sphere-Scene	Suzanne-Scene
Local AWS (Zach et al.)	0.5361	0.6310	3.0857
Local AWS (Ce Liu)	0.5767	0.6069	1.2999
TV-L^1Global AWS	0.2160	0.6363	1.1564

Table 5.3: Numerical results for the mean relative diameter errors for local and global AWS approaches with 3% image noise.

	Plane-Scene	Sphere-Scene	Suzanne-Scene
Local AWS (Zach et al.)	0.7594	0.8866	3.6669
Local AWS (Ce Liu)	1.1324	1.0174	1.7624
TV-L^1Global AWS	0.3353	0.8300	1.4680

Table 5.4: Numerical results for the mean relative diameter errors for local and global AWS approaches with 5% image noise.

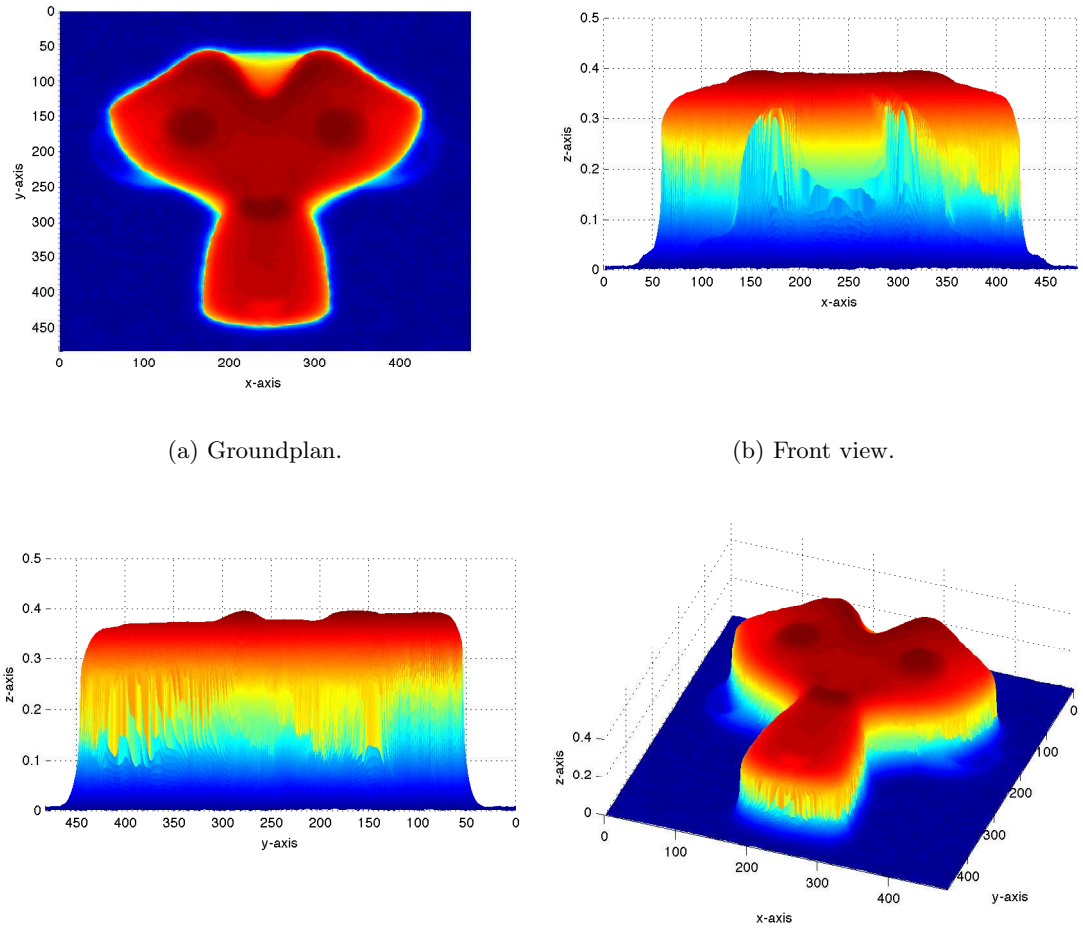
smoothness-term has been adjusted for the current aperture setting.
Figure 5.12 shows the mean diameter error for two different noise levels as well as for different numbers of aperture positions as a function of λ. The according numerical results for the mean relative diameter errors can be found in Table 5.2, 5.3, and 5.4 for images with additive Gaussian noise with a sigma of 2%, 3%, and 5% of the image dynamic range,

respectively. One recognizes, that the TV-L^1 global AWS approach reaches the best results for the Plane-Scene and Suzanne-Scene. The results from the Sphere-Scene are nearly identical for all three approaches.

The main advantage of the TV-L^1 global AWS approach is, that the smoothness-term is directly applied to the depth-map. Therefore, compared to local AWS approaches, the global version removes negative effects resulted from the least-squares circle fitting step. Moreover, the global AWS approach also allows the use of long temporal filters. Unfortunately, we could not observe a noticeable improvement due to long filter.

5.3 3D Reconstruction Results

In this section we present reconstruction results calculated using local AWS and TV-L^1 global AWS. Figures 5.14 and 5.15 show the main views and the 3D reconstruction of the Suzanne-Scene and the Sphere-Scene, respectively. The 3D reconstruction is calculated with Frigerios local AWS approach using a robust optical flow presented by Zach et al. [32]. Figures 5.16 and 5.17 show the same for the TV-L^1 global AWS approach.



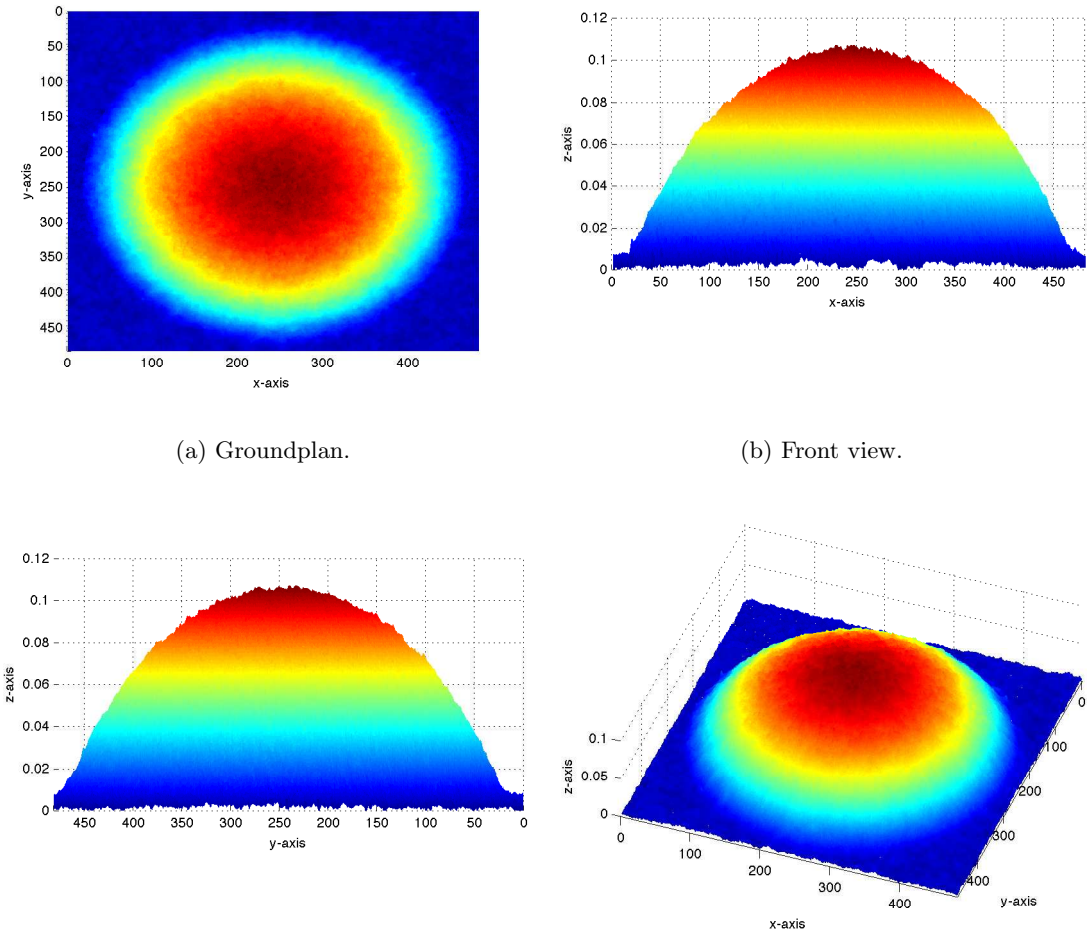
(a) Groundplan.

(b) Front view.

(c) Side view.

(d) 3D view.

Figure 5.14: 3D reconstruction and main views of the Suzanne-Scene calculated using Frigerio's Multi Image AWS algorithm with a global optical flow presented by Zach et al. [32]. (a) shows the groundplan, (b) the front view, (c) the side view, and (d) the 3D view.



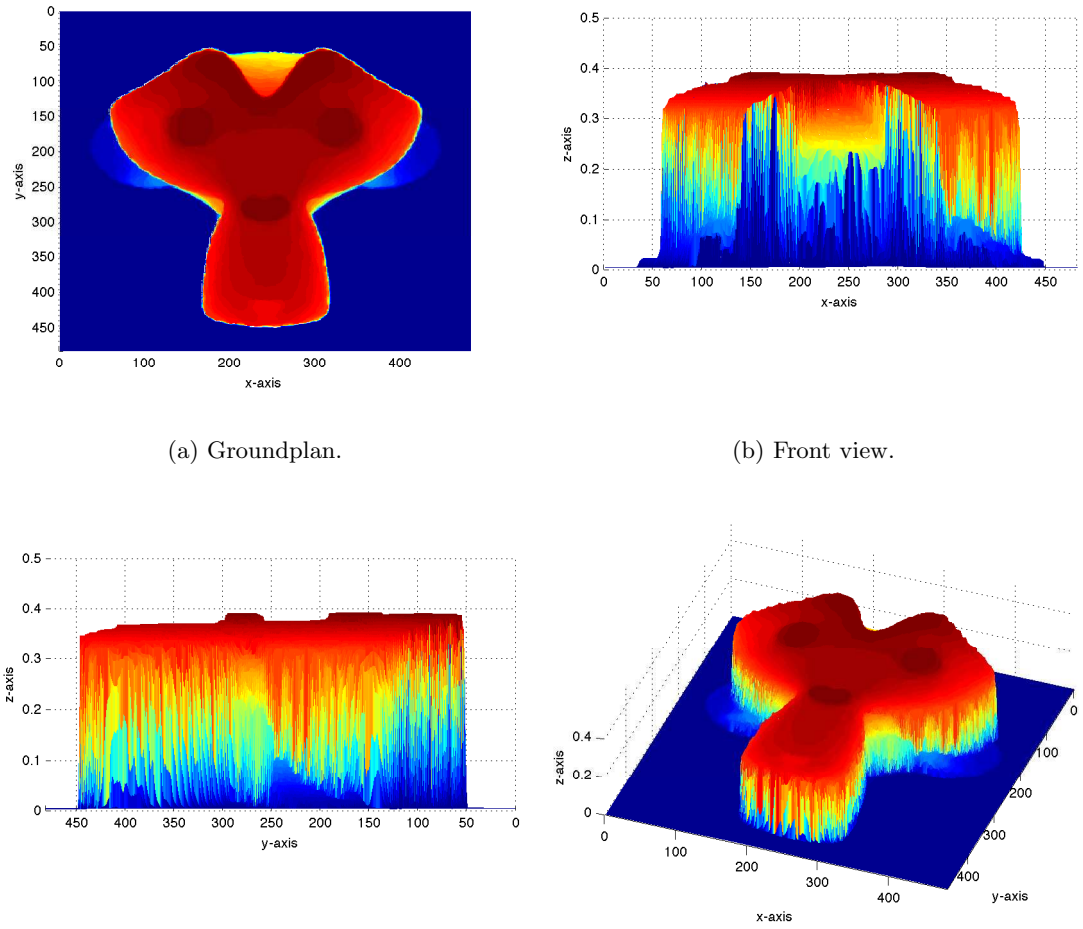
(a) Groundplan.

(b) Front view.

(c) Side view.

(d) 3D view.

Figure 5.15: 3D reconstruction and main views of the Sphere-Scene calculated using Frigerio's Multi Image AWS algorithm with a global optical flow presented by Zach et al. [32]. (a) shows the groundplan, (b) the front view, (c) the side view, and (d) the 3D view.



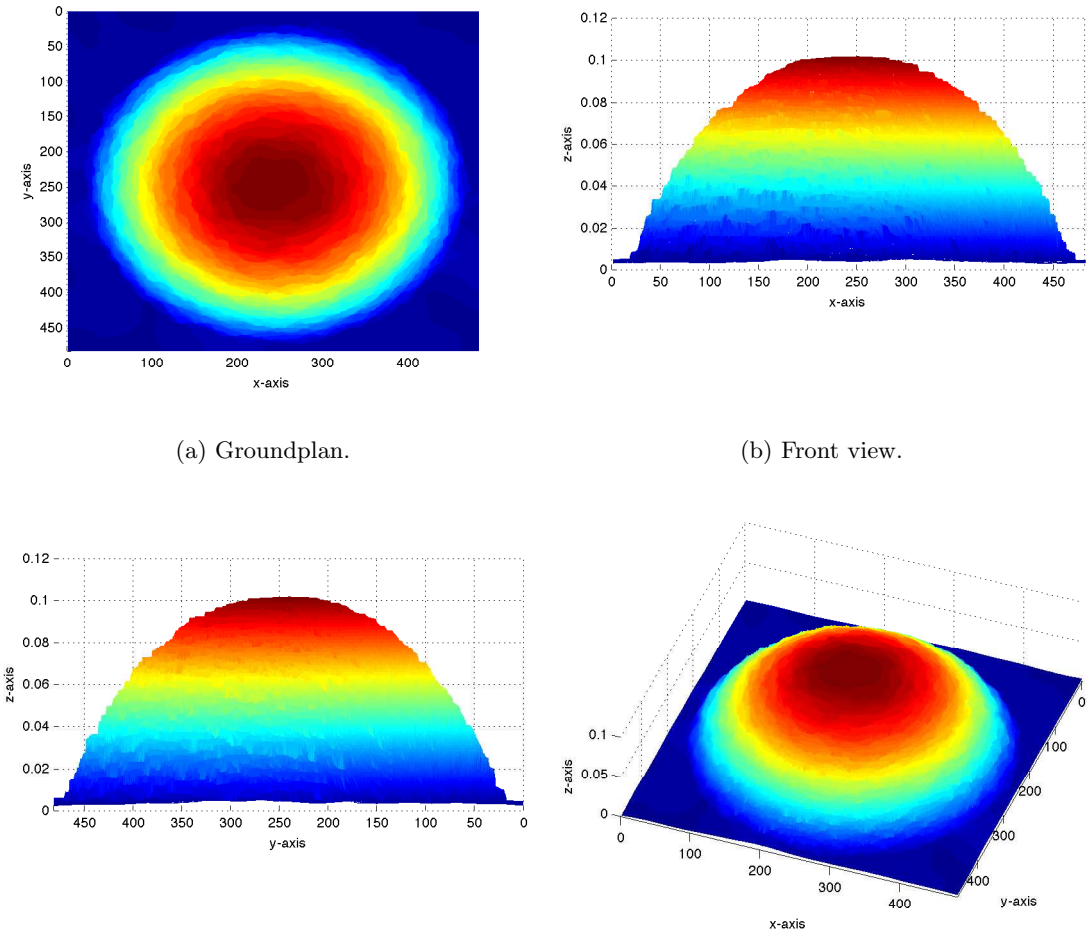
(a) Groundplan.

(b) Front view.

(c) Side view.

(d) 3D view.

Figure 5.16: 3D reconstruction and main views of the Suzanne-Scene calculated using the TV-L^1 global AWS algorithm. (a) shows the groundplan, (b) the front view, (c) the side view, and (d) the 3D view.



(a) Groundplan.

(b) Front view.

(c) Side view.

(d) 3D view.

Figure 5.17: 3D reconstruction and main views of the Sphere-Scene calculated using the TV-L^1 global AWS algorithm. (a) shows the groundplan, (b) the front view, (c) the side view, and (d) the 3D view.

Chapter 6

Conclusions

In this thesis we have evaluated a 3D surface reconstruction technique based on the Active Wavefront Sampling (AWS) approach. This final chapter gives a summary of the work and an outlook for potential future work.

In Chapter 1 we first introduced the idea of Active Wavefront Sampling (AWS) presented by Frigerio [10]. Second, advantages of AWS were outlined. It was shown, that the ability of the AWS technique to capture 3D structures using a single optical path is an attractive property. Therefore, the AWS approach has clear advantages in applications, where two separated optical channels cannot be used, like e.g. minimally invasive surgery using an endoscope, and 3D microscopy. Moreover, the AWS approach can be used to create a low cost 3D scanner, and has also the ability to calculate high accurate and robust 3D models.

Related work was presented in Chapter 2. First different 3D reconstruction approaches were described. Second, the BIRIS sensor, as an example for Static Wavefront Sampling (SWS), and the main idea of Active Wavefront Sampling (AWS) was presented.

Necessary background information was presented in Chapter 3. In this chapter topics like calculus of variations, visual motion, and optical flow were reviewed. Furthermore, this chapter also included a section regarding the theory of defocus, where the connection between a target feature's blur-circle-diameter and the target feature's depth was explained. Finally, this chapter also contained the general description of wavefront sampling, as well as a more detailed description of the multi image AWS algorithms presented by Frigerio. These algorithms were developed to use more than two sampling positions on the aperture plane.
The first algorithm defined an anchor image and performed pair-wise matching between it and the remaining images, which were acquired by rotating an off-axis aperture around the optical axis. The matching results were then used to calculate the image-rotation-diameter in a least-squares sense.
The second algorithm made use of long multi-point interpolating and derivative filter. However, the main drawback of this algorithm was the requirement of an method, that moves the spatio-temporal interrogation cuboid according to the calculated motion.

In Chapter 4 we presented the global AWS approach, which assumed that the blur-circle-radius varies smoothly almost everywhere in the image. For a better understanding, we first presented a version, which uses the L^2 norm to weight the data-term. The according minimization problem could be solved using the calculus of variations. However, due to the quadratic estimator, the L^2 global AWS approach is highly sensitive to image noise. One possible solution to solve this problem was the use of a robust estimator for the data-term, which led to the TV-L^1 global AWS approach.

Experimental results were presented in Chapter 5. Here we first evaluated the Frigerio local AWS approach by using different optical flow techniques to calculate the displacements between the images acquired form the different aperture positions. It was shown that the accuracy of the algorithm increases as the number of aperture positions increases. Furthermore, the sensitivity to Gaussian noise was evaluated. As expected, the accuracy decreases when noise increases. The image noise especially affected the methods, which make use of quadratic estimators. Moreover, we observed that the methods based on global optical flow techniques provided better accuracy results than those based on local optical flow techniques.
Next, we compared the global AWS approach to the Frigerio multi image AWS approach with long multi-point filter. It was shown, that the global AWS method creates a smoother surface and it also received better numerical results compared to Frigerio's multi image AWS approach with long filter. We showed that the TV-L^1 global AWS algorithm is robust in the presence of Gaussian image noise, and the algorithm received slightly better accuracy results than the robust local AWS algorithms. However, the main advantage of the TV-L^1 AWS approach is the fact, that the smoothness-term is directly applied to the depth-map. Therefore, compared to local AWS approaches, the global version removes negative effects resulted from the least-squares circle fitting step. Furthermore, the global AWS approach allows the use of long temporal filter.

Potential future work includes testing the approach on real-world scenes, evaluating the effect of long multi-point derivative and interpolating filter, and using other robust estimators within the data-term of the global AWS minimization problem.

Appendix A

Definitions

A.1 Abbreviations

1D	one dimension(al)
2D	two dimension(al)
3D	three dimension(al)
$\mathbb{R}^1$	Euclidean one-dimensional vector space
$\mathbb{R}^2$	Euclidean two-dimensional vector space
$\mathbb{R}^3$	Euclidean three-dimensional vector space
AWS	Active Wavefront Sampling
CAD	Computer Aided Design
CCD	Charge-Coupled Device
NCC	Normalized Cross-Correlation
PSF	Point Spread Function
SAD	Sum of Absolute Differences
SSD	Sum of Squared Differences
SWS	Static Wavefront Sampling

A.2 Used Symbols

∇	Nabla operator
Δ	Laplace operator
∂^n	derivative of order n
$\|.\|_p$	L^p norm
$*$	convolution operator
x^*	complex conjugation of x

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2:284–299, 1985.

[2] C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing, 1984.

[3] N. Asada, H. Fujiwara, and T. Matsuyama. Edge and depth from focus. 26(2):153–163, February 1998.

[4] J.-A. Beraldin, F. Blais, L. Cournoyer, M. Rioux, S. El-Hakim, R. Rodella, F. Bernier, and N. Harrison. Digital 3d imaging system for rapid response on remote sites. *3D Digital Imaging and Modeling, International Conference on*, 0:0034, 1999.

[5] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 237–252, London, UK, 1992. Springer-Verlag.

[6] P. J. Besl. Active, optical range imaging sensors. *Mach. Vision Appl.*, 1(2):127–152, 1988.

[7] S. Chaudhuri and A. Rajagopalan. *Depth from Defocus: A Real Aperture Imaging Approach*. Springer-Verlag, 1998.

[8] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(2):97–108, 1993.

[9] C. Fox. *An Introduction to the Calculus of Variations*. Dover, 1987.

[10] F. Frigerio. *3-Dimensional Surface Imaging Using Active Wavefront Sampling*. PhD thesis, Massachusetts Institute of Technology, 2006.

[11] F. Frigerio and D. P. Hart. Calibrationless aberration correction through active wavefront sampling (aws) and multi-camera imaging, 2006.

[12] I. M. Gelfand and S. V. Fomin. *Calculus of Variations*. 1963.

[13] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall, Upper Saddle River, N.J., 2008.

[14] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics. The Approach based on Influence Functions.* Wiley, 1986.

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[16] J. K. Hasegawa and C. L. Tozzi. Shape from shading with perspective projection and camera calibration. *Computers & Graphics*, 20(3):351–364, 1996.

[17] D. J. Heeger. Model for the extraction of image flow. *J. Opt. Soc. Am. A*, 4(8):1455–1471, 1987.

[18] E. Hildreth. The computation of the velocity field. In *MIT AI Memo*, 1983.

[19] B. K. Horn and B. G. Schunck. Determining optical flow. Technical report, Cambridge, MA, USA, 1980.

[20] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis.* PhD thesis, Massachusetts Institute of Technology, 2009.

[21] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. pages 674–679, 1981.

[22] H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. pages 945–951, 1983.

[23] Y. Nakagawa and S. Nayar. Shape from focus. 1989.

[24] S. K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1186–1198, 1996.

[25] M. Pollefeys, R. Koch, Luc, and L. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters. pages 90–95, 1998.

[26] M. Rioux and F. Blais. Compact three-dimensional camera for robotic applications. *J. Opt. Soc. Am. A*, 3(9):1518–1521, 1986.

[27] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[28] Y. Schechner, , Y. Y. Schechner, and N. Kiryati. Depth from defocus vs. stereo: How different really are they? In *Proc. ICPR*, pages 1784–1786, 1998.

[29] E. Simoncelli. *Distributed Representation and Analysis of Visual Motion.* PhD thesis, Massachusetts Institute of Technology, 1993.

[30] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[31] R. J. Woodham. Photometric method for determining surface orientation from multiple images. pages 513–531, 1989.

[32] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition (Proc. DAGM)*, pages 214–223, Heidelberg, Germany, 2007.

[33] E. C. Zachmanoglou and D. W. Thoe. *Introduction to Partial Differential Equations with Applications*. Dover, 1986.