Reconstruction of Intricate Surfaces from Scanning Electron Microscopy

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Jasenko Zivanov

aus Basel, Basel-Stadt

Basel, 2017

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter

Prof. Dr. Henning Stahlberg, Universität Basel, Korreferent

Basel, den 18.04.2017

Prof. Dr. Martin Spiess, Dekan

Namensnennung - Keine kommerzielle Nutzung - Keine Bearbeitung 3.0 Schweiz
(CC BY-NC-ND 3.0 CH)

Sie dürfen: Teilen — den Inhalt kopieren, verbreiten und zugänglich machen

Unter den folgenden Bedingungen:

Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.

Keine kommerzielle Nutzung — Sie dürfen diesen Inhalt nicht für kommerzielle Zwecke nutzen.

Keine Bearbeitung erlaubt — Sie dürfen diesen Inhalt nicht bearbeiten, abwandeln oder in anderer Weise verändern.

Wobei gilt:

- **Verzichtserklärung** — Jede der vorgenannten Bedingungen kann aufgehoben werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.
- **Public Domain (gemeinfreie oder nicht-schützbare Inhalte)** — Soweit das Werk, der Inhalt oder irgendein Teil davon zur Public Domain der jeweiligen Rechtsordnung gehört, wird dieser Status von der Lizenz in keiner Weise berührt.
- **Sonstige Rechte** — Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:
  - Die Rechte, die jedermann wegen der Schranken des Urheberrechts oder aufgrund gesetzlicher Erlaubnisse zustehen (in einigen Ländern als grundsätzliche Doktrin des **fair use** bekannt);
  - Die **Persönlichkeitsrechte** des Urhebers;
  - Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel für **Werbung** oder Privatsphärenschutz.
- **Hinweis** — Bei jeder Nutzung oder Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für diesen Inhalt gelten. Am einfachsten ist es, an entsprechender Stelle einen Link auf diese Seite einzubinden.

Quelle: http://creativecommons.org/licenses/by-nc-nd/3.0/ch/ Datum: 6.8.2017

Reconstruction of Intricate Surfaces from Scanning Electron Microscopy

PhD Thesis

Jasenko Zivanov

University of Basel

To see a World in a Grain of Sand
And a Heaven in a Wild Flower,
Hold Infinity in the palm of your hand
And Eternity in an hour.

William Blake

Abstract

This PhD thesis is concerned with the reconstruction of intricate shapes from scanning electron microscope (SEM) imagery. Since SEM images bear a certain resemblance to optical images, approaches developed in the wider field of computer vision can to a certain degree be applied to SEM images as well. I focus on two such approaches, namely Multiview Stereo (MVS) and Shape from Shading (SfS) and extend them to the SEM domain.

The reconstruction of intricate shapes featuring thin protrusions and sparsely textured curved areas poses a significant challenge for current MVS techniques. The MVS methods I propose are designed to deal with such surfaces in particular, while also being robust to the specific problems inherent in the SEM modality: the absence of a static illumination and the unusually high noise level. I describe two different novel MVS methods aimed at narrow-baseline and medium-baseline imaging setups respectively. Both of them build on the assumption of pixelwise photoconsistency.

In the SfS context, I propose a novel empirical reflectance model for SEM images that allows for an efficient inference of surface orientation from multiple observations. My reflectance model is able to model both secondary and backscattered electron emission under an arbitrary detector setup. I describe two additional methods of inferring shape using combinations of MVS and SfS approaches: the first builds on my medium-baseline MVS method, which assumes photoconsistency, and improves on it by estimating the surface orientation using my reflectance model. The second goes beyond photoconsistency and estimates the depths themselves using the reflectance model.

Acknowledgements

First, I would like to thank *Prof. Thomas Vetter* for his unwavering support and confidence over the years and *Prof. Henning Stahlberg* for the very insightful and encouraging discussions during that time.

Furthermore, I would like to thank *Dr. Ken Goldie* for sharing his expertise in electron microscopy, and *Dr. Martin Oeggerli*, without whom this fascinating project would not have begun in the first place.

Finally, many thanks to my friends and colleagues at the Gravis group for a very enjoyable time in a friendly and stimulating working environment.

Contents

Chapter 1

Introduction

Scanning electron microscopy (SEM) allows us to render very small objects of arbitrary topology visible to the human eye. It works by scanning the surface of a probe with an electron beam, while a nearby detector measures the electrons emitted from the surface. This results in images that look strikingly similar to optical grayscale photographs. Unlike optical microscopy, which is limited by the wavelength of visible light, SEM can resolve features as small as one nm in size. Although a number of other microscopy techniques do allow for higher resolutions, SEM is unique in its ability to make intricate microscopic shapes immediately comprehensible to the untrained human eye.

Given that the shapes depicted in SEM images can be easily understood by humans, their reconstruction by computational means has garnered only limited attention. This kind of digital reconstruction is useful for the following applications:

- **Colorization**: currently, single SEM images can be colorized by expert artists to create instructive and aesthetically pleasing pictures. Knowing the precise shape of the object allows this colorization to transferred to an entire sequence of images, yielding a colorized SEM animation.
- **Visualization**: the estimated shape can be used to compute a synthetic rendering of the object made out of any material from an arbitrary viewing angle and under arbitrary lighting conditions. This can greatly enhance the use of SEM as a teaching tool or to communicate research findings.
- **3D Printing**: additive manufacturing techniques can be used to generate a greatly enlarged copy of the original shape.

The methods that have been proposed so far aim at the reconstruction of comparatively simple surfaces.

The reconstruction of intricate shapes from images in general is still considered a challenging problem, even in the far more active field of generic *computer vision*. Researchers in that field work with all types of image-structured data, though much of the work focuses on optical images. Because the image formation process in a scanning electron microscope is in certain ways similar to that of an optical image, some of those approaches can be applied to the SEM domain. I will consider the following two areas of computer vision research in particular: *multiview stereo*, which deals with the reconstruction of 3D shapes from multiple images taken from different vantage points, and *Shape from Shading*, which attempts to reconstruct shapes from observed shading patterns, often under a known illumination.

1.1 Challenges and Opportunities

I will focus specifically on the reconstruction of intricate organic objects, such as the bodies of insects. The many thin protrusions on such objects, combined with smoothly curved untextured areas, make them a very challenging problem for current multiview stereo methods, independently of the imaging modality. Furthermore, SEM itself poses a number of additional challenges but it also provides certain opportunities.

The main SEM-specific challenge is the fact that the apparent illumination in an SEM image always rotates along with the observer. This is unavoidable, because it is only the sample that moves within the microscope, while microscope itself remains static. The vast majority of current multiview stereo methods rely on a property termed *photoconsistency*: the assumption that the same point will appear in the same color in all images. In optical images, this is equivalent to a world made out of Lambertian, i.e. perfectly matte, surfaces under a static illumination. In SEM images, however, the same point will show very different gray values when observed from different angles. This makes identifying the same point more difficult, and the problem is further exacerbated by the absence of color and by the sometimes challenging signal-to-noise ratio.

The main advantage of the SEM modality is the fact that the dependence of the lightness value on the viewing angle is very predictable. This makes SEM more amenable to shape-from-shading approaches than optical imaging. Unlike optical images, where the reflectance properties differ greatly among different materials, organic SEM samples are commonly coated with a thin layer of conductive material to increase image contrast and to prevent charging. This layer gives them an almost uniform reflectance behavior under the electron microscope. This means that the observed lightness value depends almost exclusively on the surface orientation, i.e. the direction of the surface normal at that point. As a consequence, the normal can be estimated more reliably from the observations, and it can be integrated to obtain a shape estimate.

Chapter 2

Background and Previous Work

In the following chapter, I will describe the relevant previous research directed at reconstructing shapes from images. The first part covers generic computer vision approaches, while the second part looks at SEM in particular. Because of the somewhat interdisciplinary nature of this work, this chapter is comparatively long and detailed, as it needs to introduce resarch from both of those fields. It will begin with a general overview and turn to the individual works later on.

2.1 Computer Vision

The field of computer vision strives to obtain abstract information from images. It pursues the opposite goal of computer graphics, which aims to generate images from abstract information. The applications of computer vision range from autonomous vehicle navigation and pedestrian detection, to the recognition and tracking of human faces in crowded scenes to the creation of 3D assets for the films and games industries and the digitization of historical artifacts.

Many modern methods formulate vision problems as optimization problems, where a cost or energy is minimized as a function of the unknowns. This formulation decouples the algorithm used for the optimization from the cost function itself, and it makes the specific problem formulation easier to understand. This is further amplified by the often included probabilistic interpretations, where the energy to be minimized is usually interpreted as the negative log of a likelihood, i.e. the probability of a set of parameters. This probabilistic interpretation then makes it possible to apply Bayesian statistics to those problems by weighing the observed evidence against prior assumptions.

One possible way to organize the very large number of vision problems that have been addressed over the decades would be to order them according to the amount of information that has to be extracted, which I will do in the following. This is not indended to give an overview over the entire field of computer vision, since that would be beyond the scope of this introductory chapter. The purpose of ordering the problems along the information spectrum is merely to place my own work within the larger context of computer vision research.

On the lower end of that spectrum, one would then find classification and detection problems, where an algorithm is e.g. required to ascribe one single label to a given image or a probability to a given image region. Methods that approach these problems often avoid modelling their problem domain explicitly, and instead rely on sophisticated machine learning techniques

to build highly complex models directly from labeled data. If such an algorithm is e.g. tasked with detecting pedestrians in images, it will need a reliable model that captures the very different ways in which pedestrians appear in images when seen from different sides, in different poses and under varying illumination. Because so much of the information has to be contained within the model, the representation and acquisition of that model knowledge is typically the focus of the methods on this end of the spectrum. In the case of discrete problems, such as the image labeling problem, the small amount of output information often makes it possible to explicitly evaluate all possible answers and to return the best one.

Further along the spectrum, one finds problems such as that of reconstructing the shape of an observed object that belongs to a specific known class of objects, such as human faces [1] or dolphins [2]. In this scenario, the specific shape within the class is typically represented by tens to hundreds of unknown coefficients, and the usually unknown viewing and illumination parameters represent further unknowns. Since those parameters are part of the answer that the algorithm returns, they have to be modelled explicitly. This explicit modelling of the interrelations of parameters within the model allows such models to be constructed from a smaller amount of data, since the interrelations do not need to be learned by the algorithm. Due to the increased number of unknowns, those can no longer be determined by exhaustive computation, so iterative optimization strategies are usually applied.

In most cases, the resulting problems are not convex, because the image values are themselves non-convex functions of the spatial domain. When dealing with a non-trivial non-convex problem, there is no guarantee that an iterative procedure will converge towards a global minimum. Many of the proposed algorithms thus rely on a good initialization [3, 4], and only recent methods address the problem of non-convexity through e.g. stochastic sampling [5].

On the high-information end of the spectrum, we find problems that associate at least one unknown with each pixel of an image. Those are e.g. the problems of image segmentation, where each pixel receives one discrete label, and depth estimation, where pixels receive continuous depth values.

Even further along the spectrum, we find the methods that perform such operations on voxel grids, which are equivalent to 3D images. Most importantly in the context of this thesis, here we also find the problem of surface reconstruction, which is nowadays usually posed as a segmentation problem that segments a 3D voxel grid into an inside and an outside region. This formulation guarantees a watertight surface while allowing for an arbitrary topology, limited only by the resolution of the voxel grid.

Problems on this end of the spectrum often contain millions, and sometimes billions of unknowns. Conversely, the prior assumptions about the problem can be very minimalistic, and are often limited to a smoothness assumption. Smoothness in this context means that two adjacent pixels or voxels are more likely to exhibit the same or a similar value than two very different values.

With the appropriate choice of smoothness metric, these problems can be made convex, but those metrics are also usually very strict and lead to oversmoothed results. They are thus the least appropriate for the reconstruction of intricate geometry. In addition, the large number of unknowns makes it very difficult to choose the initial state for the optimization manually.

As a consequence, modern methods that aim to reconstruct intricate geometry are rarely formulated as pure optimization problems. Instead, they more often consist of elaborate sequences of processing steps that produce a number of intermediate data terms. Those data terms are then used as parameters for a final surface reconstruction step that almost

always *is* formulated as a convex problem. In the following, I will present the previous work on the multi-view stereo problem in detail.

2.1.1 Multi-view Stereo

Multi-view stereo (MVS) refers to the reconstruction of unknown shapes from sets of calibrated images. Calibration means that functions are known that map every point in 3D space onto the image plane of each frame. The problem of finding such mappings is referred to as the Structure-from-Motion problem (SfM), and it is not the focus of this thesis. It is most often performed by matching sparse points between the images, before the shape itself is known. This type of calibration will be referred to as *geometric calibration* in this work, in order to distinguish it from the measurement of the actual brightness values, to which I will refer as *radiometric calibration*.

The MVS problem is closely related to binocular stereo, which aims to reconstruct shapes from only two images. Unlike MVS, binocular stereo does not allow for the reconstruction of complete objects, and it is usually limited to finding depth maps corresponding to the input views. A depth map is an image that contains within its pixels the distances between the observer and the respective surface points seen in that pixel.

The majority of MVS work relies on a property termed *photoconsistency*: a point in space is said to exhibit high photoconsistency if it maps onto 2D points in the different images that look similar to each other. If a point lies on the true surface of an object, then the corresponding image areas will all show the same part of that surface, and therefore they are expected to exhibit high photoconsistency. The goal is to then find a surface in 3D space that is made up of photoconsistent points, while at the same time explaining the pixels of the images.

Photoconsistency can be either measured by comparing the corresponding pixel colors directly, or by applying more abstract metrics, such as normalized cross-correlation (NCC), to corresponding image areas. The former approach is better suited for the reconstruction of small features, but it is also more susceptible to occlusions and to changes in surface radiance such as specular highlights.

The term radiance refers specifically to the amount of light emitted from the surface in the direction of the camera sensor. If that value is to remain constant over different viewing angles, then the illumination must not move with respect to the object, and the surface of the object has to be Lambertian (the latter concept will be explained in more detail in 2.1.2). In the following, I will refer to this type of photoconsistency as *strict photoconsistency*.

In context of SEM images, a more appropriate term would be electroconsistency, because no light is involved in the image formation. I will still refer to the property as photoconsistency, however, in order to maintain a uniform terminology.

More importantly, strict photoconsistency does not hold for SEM images. Instead, the illumination rotates along with the observer, because it is the sample that moves, and not the microscope. As a consequence, the gray value of a point only allows for identifying that point in images taken from similar directions. Globally, any point can theoretically appear under any gray value in every image.

The first part of this thesis will focus on methods for reconstructing intricate surfaces with small features in the absence of global photoconsistency through MVS. The second part will investigate the precise way in which the gray value in a SEM image changes as a function

of the viewing direction, and methods will be presented that extract additional information from that change.

A survey of MVS methods published before 2006 has been presented by Seitz et al. [6]. According to their taxonomy, one way to classify the different methods is based on the reconstruction algorithm that is applied:

1. Feature extraction and growing methods
2. Iterative surface evolution methods
3. Image-based methods
4. Volumetric one-shot methods

Many methods that have been presented since then combine multiple such approaches in different parts of their pipelines. In the following, I will look at each of them with a view towards the reconstruction of intricate geometry from SEM images.

1. Feature Extraction and Growing Methods

These methods work on sparse points and are therefore inadequate for the reconstruction of intricate geometry. The today probably most prominent representative of these is PMVS [7]. This method estimates a set of sparse planar patches in space and fits their orientation to the input images.

The resulting patches are equivalent to a cloud of oriented surface points and are often used to compute a watertight surface by one of the volumetric surface reconstruction methods that are described further below.

2. Iterative Surface Evolution Methods

Here, an initial surface estimate is iteratively optimized according to some cost measure in order to better fit the input images. The main advantage of these methods is their ability to model occlusion geometrically, since the visibility of individual points in space can be determined from the current shape estimate. This can also lead to errors in cases where that estimate is wrong.

An early such method is voxel carving [8], where the initial surface contains the entire object, and voxels on the surface are iteratively removed if they exhibit insufficient photoconsistency. This leads to surfaces that balloon outward in the smooth areas, and it can also damage surfaces in the case of specular highlights. The inability of that algorithm and its variants to un-carve voxels that have been removed compromises their stability.

This particular problem was solved through level set methods [9, 10, 11]. Here, an arbitrary surface, represented by a level set in a 3D scalar field, is evolved in both directions (i.e. material is added or removed) to increase photoconsistency and to decrease an additional regularization energy. These methods are still local, so they rely on an appropriate initialization. This is a problem particularly with respect to the visibility estimation.

Although not a surface evolution technique, the recently proposed inverse ray-tracing method by Liu et al. [12] is also a local method, since it performs loopy belief propagation on a Markov random field (MRF). The main difference to surface evolution techniques lies in the fact that the scalar field is evolved everywhere in space at the same time, and not only along

the current surface estimate. Since it considers the pixels individually, this method is able to reconstruct very complex geometry showing complicated self-occlusion. It is, however, computationally very demanding, since the MRF formulation used considers cliques comprising up to thousands of random variables. It is also dependent on strict photoconsistency.

3. Image-based Methods

In image-based methods, the scene is described by a grid of depth values, usually coinciding with the pixels of one of the input images. These depth images are referred to as depth maps. This representation is nowadays mostly used when time is an essential factor, like e.g. in autonomous navigation [13]. In that case, the aim is not the precise reconstruction of intrictate geometry. Another application of depth maps is the reconstruction of scenes that span too many different levels of scale to be effectively represented by a voxel grid [14, 15, 16]. In those works, the depth maps are not the final result of the algorithm, but are used to construct a mesh using volumetric Delaunay triangulation. The depth maps are estimated using normalized cross-correlation (NCC) of image windows, SIFT [17] or PatchMatch [18] descriptors. Since those windows and descriptors carry information collected from image areas of a certain size, they cannot adequately describe features smaller than that size.

A notable exception is the recent method by Kim et al. [19] which works on very dense image sequences and aims to estimate the depths of individual pixels independently. Although it can reconstruct very thin features, it assumes strict photoconsistency and clearly distinguishable colors, which prevents its application to SEM images.

In spite of the small number of purely image-based methods that have been published in recent years, recent work on volumetric methods [20, 21] has shown that accurate depth maps can greatly improve their performance. In chapter 4, I will present a depth estimation technique that works on individual pixels and that can cope with noisy gray-level images, as long as the images are taken from sufficiently close view angles so that the gray values of given points do not change excessively. The key to that method is that it computes both the depth value and a denoised gray value simultaneously.

4. Volumetric One-shot Methods

Volumetric one-shot methods have become very popular over the past decade. They first use the images to compute local energy terms defined on a voxel grid, and then they extract a 3D surface that is optimal under those terms. In many cases, a scalar regional term expresses a preference of a voxel for being labelled object or empty space, while a surface energy term describes the likelihood of a surface traversing that point in space. The final surface extraction is then equivalent to the computation of the most likely inside-outside segmentation of 3D space. Unlike in the case of surface evolution methods, the computation of this segmentation is usually a convex problem with a unique solution that does not depend on the initialization. In almost all cases, a mesh is finally extracted from the segmentation using the marching cubes algorithm [22]. The methods that I have developed as part of this thesis fall into this fourth category if considered in their entirety.

Because methods of this type cannot rely on geometric visibility information in the way iterative surface evolution methods can, they need to estimate image correspondence in an occlusion-robust way. All of the methods surveyed below that estimate image correspondence

(and thus depth) do so by correlating image windows of a certain size using normalized cross-correlation (NCC). The advantage of this metric is the low probability of observing a strong correlation accidentally. Even though this can happen if only two views are considered, particularly if the scene contains periodic patterns, it is very unlikely that those mismatches would coincide in space for multiple image pairs.

The use of NCC as a measure of photoconsistency allows these methods to essentially count the number of images where a given depth shows strong correlation to the reference image. Images in which that point is occluded do not contribute to the total matching score, but they are also not likely to corrupt the correct depth. The main drawback of NCC as a similarity metric is the fact that it is measured for an entire window worth of pixels simultaneously. This leads to artifacts in cases where a feature is too small to be entirely covered by a window.

While many surface reconstruction techniques have been proposed as part of full MVS pipelines, others have been presented as independent methods. In those cases, the methods merely assume that adequate depth maps or point clouds are available, and they make no distinction whether these have been obtained through MVS or by other means, such as laser-range or structured-light scanning. In both scenarios, the final volumetric segmentation makes it possible to remove noise and outliers from the initial measurements through a process equivalent to local probabilistic reasoning within the voxels of the volume.

In the following, I will discuss both types of surface reconstruction together, because some of them are very closely related, even though certain variants contain a depth estimation step while others do not.

Volumetric Surface Reconstruction

Three main strands of volumetric surface reconstruction have emerged over the years: discrete Markov-random-field (MRF) based methods, total-variation (TV) based convex relaxation methods and Poisson surface reconstruction. All three approaches aim to estimate a scalar indicator function $u(x)$ that is equal to one inside the object and zero outside. They all suffer from a minimal surface bias, since the cost of a surface must always be positive to keep the problem well-posed. This minimal surface bias tends to cut off thin protrusions and to fill in cavities.

MRF-based methods: The MRF-based methods [23, 24, 25, 26] formulate the problem in a discrete way, by defining a graph that consists of the voxels as nodes while the edges between them are given by pairwise neigborhood relations. Each voxel is associated with a scalar *unary term* that defines the cost of the voxel being labelled either inside or outside, while each edge carries a *binary term*, defining the cost incurred if the two attached voxels do not share the same label. This binary term corresponds to the local cost of a surface. This formulation is equivalent to the Ising model of ferromagnetism, which has been studied for almost a century at the time of this writing. As long as all binary terms are non-negative, a globally optimal segmentation can be computed using the min-cut algorithm [23].

The main disadvantage of this approach is the discrete problem structure. The total surface cost is equal to the sum of the binary terms of all edges that coincide with inside-outside transitions. As a consequence, a diagonal surface can be up to $\sqrt{3}$ times more expensive than an axis-aligned one. Although this can be alleviated to some degree through the use of a more complex graph structure, i.e. edges to more than the six immediate neighbor voxels, this also greatly increases the computational complexity of the problem.

The very early MRF-based MVS method proposed by Vogiatzis et al. in 2005 [24] is noteworthy in that it avoids computing image depths altogether as an intermediate step. Instead, the binary terms are given by a photoconsistency measure that is evaluated at every voxel. If a point in space projects onto image areas that appear similar to each other, then the surface cost is lower in that area of space. As a photoconsistency measure, the authors apply the normalized cross-correlation (NCC) between image patches of a certain size centered around the point in question. The unary terms require points on the boundary of the volume to be labelled outside, while points sufficiently deep within the scene are always labelled inside. An additional heuristical inflationary term favors the voxels in between being labelled inside. In spite of this latter term, the method still tends to cut off thin protrusions, because those require a large surface area, while the cost of mislabelling the small number of voxels contained inside is comparatively low.

The same authors have later improved on their algorithm [25] by introducing a robust depth voting scheme. In this formulation, the depth of maximal photoconsistency is first determined for each pixel of each image. The binary terms are then only reduced for edges that fall close to those optimal depths. This helps to sharpen the binary terms and leads to a better reconstruction of corners and sharp edges, but the central problem of thin protrusions remains. This estimation of an optimal per-pixel depth is equivalent to the computation of a depth map.

Also worth mentioning is the early method by Sinha and Pollefeys [27] that aims to enforce precise silhouette consistency in addition to photoconsistency. Most of the methods mentionend above also consider silhouette information in a negative sense, i.e. any point in space that projects outside the silhouette in *any* of the images is required to be classified as outside. In contrast, the method by Sinha and Pollefeys also requires every image point within the silhouette to back-project onto an object surface and not the background. This is accomplished by discretizing 3D space in such a way that silhouette consistency can be formulated as a hard constraint. The idea of enforcing strict silhouette consistency would appear again in the context of TV-based continuous surface reconstruction.

Poisson reconstruction methods: While the discrete MRF-based methods work on a graph that represents the scene, the continuous Poisson reconstruction methods aim to estimate a scalar indicator function $u : \mathbb{R}^3 \mapsto \mathbb{R}$ that maximizes consistency with a discrete set of surface normals n_i known at certain points, while at the same time minimizing an L^2 regularization energy, $|\nabla u(x)|^2$. This is accomplished via the minimization of the functional $|n(x) - \nabla u(x)|^2$, where $n(x)$ is zero everywhere except at the given discrete points. The Euler-Lagrange equation associated with this energy functional is the Poisson equation $\Delta u(x) = \mathrm{div}(n(x))$, hence the name.

A precursor to this family of methods was proposed by Davis et al. in 2002 [28], where the authors aim to fill gaps in given 3D meshes through linear diffusion on a regular voxel grid. The input information takes the form of boundary conditions, i.e. certain voxels around the mesh geometry are always defined as either inside or outside (i.e. $u(x) = \pm 1$), and that information is propagated into the remainder of the volume though diffusion. A linear diffusion process, $\dot{u} = \mathrm{div}(\nabla u(x))$, corresponds to a gradient descent in the aforementioned L^2 regularizer.

The term *Poisson reconstruction* was coined only later [29], where the problem takes the form common today of fitting a function $u(x)$ to a cloud of oriented points. In that formulation, the input points no longer constitute boundary conditions but are instead contained in the

vector field $n(x)$, as described above. This formulation allows the input points to contain a certain amount of noise, and the method interpolates gracefully between them.

This and later works have shown that the problem can be solved very efficiently using an adaptive octree representation [29], parallelizable multigrid techniques [30] and even pure GPU implementations [31]. Originally, the unknown integration constant was estimated as a constant global value [29]. The approach was later made more robust by allowing the integration constant to vary smoothly across space as well [32]. This variant is known as *screened* Poisson reconstruction.

Traditionally, Poisson reconstruction methods were formulated as pure surface reconstruction techniques that take a cloud of oriented points as input. Since no information is available at all in areas far away from those points, this tends to lead to surfaces that balloon into those empty areas. Shan et al. [20] could improve dramatically on those results by constructing dense depth maps that correspond to the input images and that exhibit depth discontinuities coinciding with the edges in the images. These contour-correct depth maps are then used to augment the Poisson reconstruction approach through the addition of free-space voting, i.e. a term that encourages areas seen in front of observed points to be classified as empty space. This allows their algorithm to use the depth maps as a local silhouette constraint, leading to considerably better reconstructions of the internal (i.e. non-silhouette) contours of the object. This latter work clearly illustrates the need for contour-correct dense depth maps in MVS, even if they are not the final result of a pipeline.

TV-based convex relaxation methods: Like the Poisson reconstruction methods, these methods are also continuous, and they also aim to find an optimal indicator function $u(x) : \mathbb{R}^3 \mapsto [0, 1]$. The main difference is that they use an integral over the L^1-norm of the gradient, $|\nabla u(x)|$, as a regularizer, i.e. the total variation (TV). Unlike the L^2 regularizer which always prefers a smoother function, the L^1 regularizer is better suited for the reconstruction of piecewise constant functions. This can be illustrated using a minimalistic discrete 1D example.

Let x_1, x_2, and x_3 be three equidistant neighboring points along a 1D line, and let the function values for the two outer points be fixed, $f(x_1) = a$ and $f(x_3) = b$. If we aim to optimize the value of $y = f(x_2)$ under an L^2-regularizer, then the total energy will be equal to $(y-a)^2 + (y-b)^2$. The minimum of that energy is given at $y = (a+b)/2$, i.e. the mean of the two points. If we look at the L^1 energy instead, then that energy, $|y-a| + |y-b|$, will be constant and equal to $|a-b|$ for all values of y between a and b. The L^1 energy is indifferent to the precise value of y, as long as it is located between the two sample values. It can thus tolerate arbitrarily sharp edges, while the L^2 energy always prefers a smooth solution. This effect forms the core of the seminal denoising model by Rudin, Osher and Fatemi [33] from 1992 and its extension by Chan and Esedoglu [34].

Another interesting property of the TV regularizer comes into play when it is used in the solution of a segmentation problem, such as surface reconstruction. In that case, the TV integral is equivalent to the perimeter of the enclosed set, i.e. the surface area. While the discrete MRF-based methods only approximate the surface area by the number of inside-outside transition edges, the TV-based convex relaxation methods aim to minimize the actual surface area.

Analogously to the unary terms in the MRF scenario, and unlike the classical point-cloud based Poisson methods, each voxel carries a real-valued scalar parameter that biases that voxel towards preferring to belong to either the inside or the outside partition of the volume.

This parameter is usually referred to as a *regional term* in this context. While the TV approach avoids the discretization errors of the MRF approach and allows for a more memory-efficient optimization, the essential difficulties of reconstructing thin protrusions remain.

An early such method was the depth fusion technique presented by Zach et al. in 2007 [35]. There, in addition to the homogeneous TV model, the authors also propose a weighted TV model, building on the active-contour based image segmentation method by Bresson et al. [36]. In the weighted TV model, the TV integrand $|\nabla u(x)|$ is replaced by a locally weighted one, $g(x)|\nabla u(x)|$. The scalar function g assumes the role of the binary terms from the MRF formulation, and it encourages the surface to pass through points where g is small.

The TV approach was later reformulated by Kolev et al. [37, 38] as a full MVS method, using continuous adaptations of the unary terms proposed by Vogiatzis et al. [24, 25] as regional terms. Another paper by Cremers and Kolev [39] focuses on enforcing precise silhouette constraints in addition to consistency with depth estimates derived from MVS. This is shown to greatly improve the reconstruction of thin features, as long as they are silhouetted against the background in some of the images. Unlike the method by Sinha and Pollefeys [27], this is not accomplished through an irregular volume discretization, but instead by iteratively projecting the resulting surface onto the most similar one that fits the silhouettes.

A later paper by Kolev et al. [40] reformulates silhouette consistency in an exact probabilistic way, but it abandons the idea of strict silhouette consistency and it treats silhouette information as uncertain instead. The probability of a pixel belonging to either foreground or background is given by two color distributions that are measured from the images.

Although silhouette information helps greatly in the reconstruction of thin features, it can only be exploited if the scene can be trivially segmented into a foreground and a background. In the SEM setting, both areas consist of the same gray values and the edge between them can be arbitrarily faint, so silhouette information is generally not available. In addition, features located within concavities can never be seen silhouetted against the background.

In a different paper, Kolev et al. [41] have shown that the reconstruction of thin structures can also be improved by making the surface cost anisotropic, i.e. dependent on the orientation of the surface. Formally, this is accomplished by minimizing $|D(x)\nabla u(x)|$ instead of $g(x)|\nabla u(x)|$, where $D(x)$ is a regularization tensor that takes the form of a symmetric, positive definite 3×3 matrix.

Here, D behaves similarly to the diffusion tensor under anisotropic diffusion. A process of anisotropic diffusion, $\dot{u} = \operatorname{div}(D(x)\nabla u(x))$, is indeed equivalent to a gradient descent in the corresponding L^2 energy, $|D(x)\nabla u(x)|^2$. The eigenvectors of D form an orthogonal system, and the amount of diffusion along each of those three directions is proportional to the corresponding eigenvalue [42, 43, 44]. When used as a regularizer, the eigenvalues of D determine the cost of a surface running *orthogonally* to their corresponding eigenvectors.

The method by Kolev et al. [41] always assumes a locally planar surface, so the regularization tensor $D(x)$ exhibits one small eigenvalue at most, while the other two or three are large. This results in a regularizer that allows for only one surface orientation in any given area of space. That orientation is taken from planar PMVS patches [7] that can only provide reliable orientation estimates for textured, locally planar surfaces. While this type of planar anisotropy indeed helps in the reconstruction of thin disc-shaped features, it tends to destroy thin cylindrical features.

This problem is addressed by the anisotropic depth fusion method by Schrörs et al. [45], where the authors allow for all four possible types of eigenvalue configurations, corresponding to

corners (three small eigenvalues), sharp edges or ridges (two small eigenvalues, one large), planes (one small, two large) and homogeneous regions (three large eigenvalues). In this approach, the regularization tensor is computed directly from the current estimate of u.

In chapter 6, I will present a regularizer that extends on this idea by estimating the local structure directly from the images. Specifically, this novel approach abandons the assumption that a local surface normal can even be known in all circumstances. Instead, I argue that a set of observations of the same edge from multiple views only allows for determining one of the two dimensions of the normal around that edge. The complete normal can only be estimated by considering multiple edges in close proximity.

2.1.2 Shape from Shading and Photometric Stereo

The term Shape from Shading (SfS) refers to the problem of finding a 3D surface that explains the smooth radiance changes observed in *one single image* that stem from changes in *surface orientation*. This excludes radiance changes resulting from observed contours or cast shadows. Since only one single image is used, the surface can be represented by a depth map, $u(x, y)$.

The first application of such a method known to me was proposed by Rindfleisch [46] to the reconstruction of lunar topography along parallels (i.e. lines of constant latitude) in 1966. The term shape from shading itself was only coined by Horn in 1970 [47]. There, he already suggests an application of the approach to shape reconstruction from secondary-electron SEM images, though the shading model he applies does not consider the position or shape of the detector and it does not account for cast shadows. These effects and the different types of SEM images will be discussed in section 2.2.

In 1977, Horn proposed the concept of a *reflectance map* $R(n)$ for distant illumination environments viewed under an orthographic projection [48]. $R(n)$ is a 2D scalar field that maps depth gradients ∇u (that are equivalent to surface normals $n \in \mathbb{S}^2$) onto the radiances $v \in [0, \infty[$, that a point will emit if it exhibits normal n. The reflectance map thus encapsulates both the distant illumination and the reflectance properties of the object.

Also in 1977, Nicodemus et al. proposed the bidirectional reflectance distribution function (BRDF) [49] as a property particular to a given material. It takes the form of a 4D scalar field $f_\lambda(\omega_i, \omega_e) : \mathbb{S}^2 \times \mathbb{S}^2 \mapsto \mathbb{R}^+$ that, for each given wavelength λ, describes the amount of light emitted in a given direction ω_e when the surface is irradiated by light coming from a given incidence direction ω_i. Both directions ω are given relative to the surface normal, which we can define as $(0, 0, 1)$ without loss of generality.

From the linearity of light that was first noted by J. H. Lambert in his *Photometria* in 1760 [50], it follows that Horn's reflectance maps are integrals over the illumination environment $L(\omega)$ weighted by the BRDF of the observed material:

$$R_\lambda(N) = \int_\Omega \max(0, \omega \cdot n) L(\omega) f_\lambda(A_n\omega, A_n\omega_e) d\omega, \tag{2.1}$$

where A_n are orthogonal 3×3 matrices that rotate the surface normal n into $(0, 0, 1)$, and ω_e is the reverse viewing direction that is constant under an orthographic projection. The factor $\max(0, \omega \cdot n)$ accounts for the fact that the irradiance of a surface element dA is proportional to the surface area that dA assumes from the point of view of the light source [50]. Please note that above definition is only unique for isotropically reflecting materials, because the matrices A_n are only specified up to a final rotation around the z-axis. The radiance of

anisotropically scattering materials, such as brushed metal, feathers or fur, are not constant under such rotations. I will, however, only consider isotropical reflectance in this thesis.

Furthermore, the above definition assumes that the illumination field $L(\omega)$ is homogeneous across the volume. This assumption only holds for convex surfaces, because certain incoming light directions can be occluded within cavities, which produces shadows. If L consists of a single sharp peak, e.g. $L(\omega) = \delta(\omega - \omega_L)$, then we speak of a directional light, or a distant point light. Under that type of illumination, points where that peak direction is occluded will lie in a shadow. If the illumination is a wider function, then we speak of soft lighting which casts soft shadows. In that case, more points will be affected by shadows, but fewer of them will be completely dark. The extreme case of this is a uniform function $L(\omega) = L_0 \in \mathbb{R}^+$, which is approximated by the illumination on a foggy or cloudy day.

The simplest BRDF is constant, and such a material is said to exhibit *Lambertian* reflectance. The value of that constant is referred to as the *albedo* of the material.

Two distinct types of reflections occur in nature: diffuse and specular reflections. A diffuse reflection is observed when the incoming light raises the electrons within the material into excited states. When they leave those states, they emit the very specific energy difference in the form of a photon that is released in a random direction, with a wavelength that corresponds to the energy difference. Due to the randomness of the direction, the behavior of diffuse reflectors is often near-Lambertian, i.e. it does not depend strongly on the position of the observer. Deviations from Lambertian behavior do occur, however, because the surface normal is never constant across the surface area covered by a pixel. Due to microscopic shadowing and masking effects, rough surfaces generally scatter more light back in the incident direction than in other directions. This behavior is modelled explicitly by the Oren-Nayar shading model [51].

Specular reflections are wave effects that happen at the surface of the material. When an electromagnetic wave traverses the interface between media of different optical density, a part of the wave is reflected back. The reflected wave travels in the incoming direction mirrored on the surface normal. For a perfectly reflecting mirror surface, the BRDF is given by $f(\omega_i, \omega_e) = \delta(\omega_e - (2\omega_i \cdot n - 1)\omega_i)$. The local distribution of surface normals under a pixel generally blurs this mirror reflection, leading to a wider peak around the mirror direction. A very prominent effect for specular reflectors, especially dielectric ones, is the Fresnel effect. It produces specular reflections that are much stronger at grazing angles (i.e. $\omega \cdot n$ is small) than at more frontal angles.

In 1979, Woodham proposed the technique of **photometric stereo (PS)** [52] that entails the reconstruction of a shape from multiple images of the *same* object seen from the *same* point of view under *different* illuminations. In order to distinguish PS from the binocular and multiview stereo methods discussed in 2.1.1, I will refer to the latter as *photogrammetric* stereo methods in the following.

Although PS is essentially an extension of SfS, the two concepts have been considered separately in literature ever since. This is because SfS aims to estimate a 2D quantity, the surface normal $n \in \mathbb{S}^2$, from a single radiance value $v \in \mathbb{R}$. As this is an underconstrained problem (with the exception of certain singular points), SfS is unable to estimate the normal locally. Instead, the normals of all points have to be estimated simultaneously. As soon as observations under two or more illumination environments are available, the normals of the individual pixels can be determined separately, at least up to pointwise ambiguities. Then, surface reconstruction reduces to a problem of numerical integration from noisy gradients.

The methods I have developed as part of this thesis would be properly considered PS meth-

ods, and not SfS. Overviews of later pure SfS methods can be found in the surveys by Zhang et al. from 1999 [53] and by Durou et al. from 2007 [54]. A more recent survey of PS methods was published by Herbort and Wöhler in 2011 [55].

An interesting property of SfS and PS methods that had already been noted by Horn [47] is their complementarity to photogrammetric stereo methods. While photogrammetric stereo allows us to estimate the depth of sharp edges where SfS methods often fail, SfS allows us to estimate the depth of smooth regions where this is not possible for photogrammetric stereo approaches. In 1983 and 1987, Ikeuchi developed the first algorithms based on this idea [56, 57], by fusing surface orientations obtained through PS with sparse depth maps obtained by binocular stereo.

A slightly different type of complementarity was described by Nehab et al. in 2005 [58]. There, the authors look at the problem in frequency space, and they observe that the integration of noisy surface orientations obtained through PS leads to a degradation of the low frequencies of the depth function. At the same time most triangulation-based methods (e.g. both photogrammetric stereo and active methods such as structured-light or laser-range scanning) reconstruct the low frequencies well, but they miss the high-frequency components instead. The authors then present an algorithm that allows them to fuse the two types of information efficiently. Their triangulated depths are determined through a structured-light scanning setup. In 2009, I proposed a similar approach [59] for the reconstruction of human faces, where the high-frequency component is measured through PS, while the low-frequency component stems from a fit of a statistical shape model.

In 2006 and 2008, Hernandez, Vogiatzis and Cipolla [60, 61] have proposed combined MVS/PS methods that aim to reconstruct very smooth, shiny and untextured surfaces from multiple views under a point light illumination. Their capture setup allows for a reliable foreground/background segmentation, which provides strong silhouette constraints to their algorithm.

This concludes my review of the relevant computer vision methods. In the next section, I will discuss their applications to the SEM domain.

2.2 Scanning Electron Microscopy

In the following, I will first explain the way in which an image is formed in a scanning electron microscope, since this is necessary in order to discuss the existing methods. The discussion of those methods will be given afterwards.

2.2.1 Image Formation

As mentioned in the introduction, a scanning electron microscope generates an image by scanning the scene with a focused electron beam. Every pixel corresponds to a particular beam direction, and the pixels are recorded sequentially. When the electron beam strikes the surface, electrons are emitted which are then captured by specialized detectors located nearby.

Since every pixel corresponds to an electron ray, all of which originate from the same source (the final aperture of the objective lens), it is this origin of the electrons that corresponds to the eye in an optical image. This means that the particles travel in the opposite direction compared to optical imaging.

When the electron beam collides with the surface, different types of electrons are emitted: most importantly, the slow secondary electrons (SE) and the faster back-scattered electrons (BSE). Specialized types of detectors are used to capture those two types of electrons, and they can be used simultaneously. The number of emitted electrons depends on the energy of the scanning beam, the material being scanned and on the angle between the beam and the surface normal at the point of impact. The distribution of exitant directions depends on the angle of incidence as well. Just like in the case of optical imaging, this behavior can be encapsulated by a BRDF, although the specific functions differ.

Once an electron is emitted from the surface, it still needs to be captured by a detector. Whether that happens depends on the direction of travel of the electron and on the location and shape of the detector. Since the value that is finally stored in a given pixel is proportional to the number of electrons captured during the time interval corresponding to that pixel, it is the detector that corresponds to the light source in an optical image.

Even if the electron is emitted in a direction that points to a detector, it can still be re-absorbed by surrounding matter. This occlusion effect corresponds to shadows in optical images. Detectors for secondary electrons are usually surrounded by a charged grid. This serves to attract electrons that would otherwise miss the detector, and thereby boosts the effective signal-to-noise ratio. A greater detector charge thus corresponds to an increase in the effective size of the detector. Analogously to optical images, a larger effective detector size corresponds to a larger light source and thus to softer illumination and softer shadows.

Qualitatively, the behaviors of the two mentioned types of electrons differ as follows. Secondary electrons are generated within an area termed the *interaction volume*, which is located beneath the impact point. If the angle between the beam direction and the normal is large, i.e. if the surface exhibits a significant slope, then the interaction volume is more exposed and a greater number of SE is emitted. This leads to an edge highlighting effect, and it is qualitatively similar to the Fresnel effect of specular reflections in optical images. This is possibly the main reason why SE images are immediately comprehensible to untrained humans, though this assumption would require further examination. SE images exhibit very soft shadows, similar to those seen under uniform optical illumination. The total number of SE captured is generally greater than that of BSE, so SE images are less noisy.

Back-scattered electrons penetrate deeper into the material, and their intensity depends more strongly on the composition of that material. Specifically, materials containing heavier atoms will produce a greater number of BSE. As their name implies, BSE are mostly scattered in the direction of beam incidence. For that reason, BSE detectors are usually mounted around the objective lens and exhibit a ring-like shape. This ring is in some cases separated into a number of segments, and the numbers of electrons captured by each of those ring-segments can be read out separately. BSE are usually much faster than SE and travel along mostly straight lines, so their shadows are much harder. They are similar to the shadows cast by a ring-shaped lightsource, such as the ring-shaped lamps that would be found around a cosmetic mirror. If only a segment of the ring-shaped detector is used, then the shadows appear similar to those cast by an elongated light source.

2.2.2 Shape Reconstruction from SEM images

The photogrammetric approach to shape reconstruction from SEM images has been discussed by Piazzesi in 1973 [62], where he presents simplified photogrammetric equations that arise under one tilt-axis in the SEM scenario. Since the matching of surface points does not differ from the same process in optical stereo methods, the optical methods are usually applied as-is

[63, 64, 65]. In all surveyed papers, this matching was performed either through comparisons of image windows or through the use of high-level descriptors such as SIFT [17]. As discussed in 2.1.1, this does not allow for the reconstruction of fine-scale surface details.

The reconstruction based on SfS and PS depends on the assumed reflectance maps. The early SfS work by Horn from 1970 [47] assumes that SE reflectance can be approximated by an inverse cosine law,

$$v \approx n_z^{-1}, \tag{2.2}$$

where n is the surface normal and n_z its component in (reverse) beam incidence direction. In 1981, Ikeuchi and Horn applied [66] an updated reflectance function,

$$v \approx (1 + n_z^{-1})/2. \tag{2.3}$$

The special case of two symmetrically mounted detectors on opposite sides of the scene has received particular attention over the years. For two such BSE detectors, Lebiedzik has established [67] the empirical relation,

$$\sin(i) = \frac{n_x}{n_x^2 + n_z^2} \approx \frac{v_R - v_L}{v_R + v_L}, \tag{2.4}$$

where n is the unit surface normal, i the lateral inclination angle and v_R and v_L are the detector responses from two BSE detectors mounted on the left and right side of the scene. The approximation is valid for angles $|i| < 60°$, and robust to variations in beam intensity and material composition. An analogous relation for SE was proposed by Reimer and Stelter in 1987 [68],

$$\sin(\phi)\sin(A) = n_x \approx \frac{v_R - v_L}{v_R + v_L}, \tag{2.5}$$

where the azimuth angle A is the angle between the projections of the detector direction e_x and the normal n onto the frontal XY plane, and the inclination angle ϕ is measured relative to the beam direction e_z. This model can be derived from the assumption of an inverse cosine emission yield in conjunction with a Lambertian reflectance (i.e. uniform distribution over the emission directions and Lambert's cosine law) and infinitely small detectors.

In 1991, Beil and Carlsen proposed a combined binocular-stereo/PS algorithm [69] that uses both of these relations for symmetrical detector arrangements. This method applies the framework proposed by Ikeuchi in 1987 [57] which relies on a coarse-to-fine strategy for stereo matching. As has been noted more recently in the optical context [19], such a strategy is unable to deal with thin features.

These symmetrical arrangements were studied in more detail by Vynnyk et al. in 2010 [70], resulting in a more advanced raflectance map for SE. Their model considers the absorption of electrons by the electron gun, local self-shadowing (i.e. *not* cast shadows) and it represents the relation between the normal and the two detector responses by a non-monotonic function. The latter fact prevents the model from reconstructing normals that form an angle of more than 45° with z, the beam incidence direction.

The symmetric two-detector arrangement can be extended to four detectors, which allows for a more stable reconstruction, even from only one view [71]. Such a four-detector system has also been simulated by using one detector and rotating the probe four times by 90° around the z-axis [72].

All of those methods rely on a symmetrical detector arrangement, so they are only applicable if the corresponding equipment is available. The recent method for silicon wafer verfication

by Estellers et al. [73] performs SfS from only one image, and it applies prior knowledge in the form of a deformable template. The reflectance model used is the inverse cosine model that had already been applied by Horn [47]. The method by Danzl and Scherer from 2001 [74] is the only one that has come to my attention that aims to estimate the reflectance map ad hoc, in conjunction with photogrammetric stereo information. Their reflectance model is a free fourth degree polynomial of the angle between the beam direction and the surface normal. This definition prevents it from considering the position of the detector, leading to reflectance maps that are always radially symmetrical. None of the surveyed methods that consider shading information aim to reconstruct a full 3D shape. Instead, they all work on depth maps exclusively.

2.3 Contributions

My thesis makes the following contributions to shape reconstruction from multiple SEM images based on MVS alone as well as MVS in combination with PS. The first part of the thesis deals with photogrammetric reconstruction exclusively, i.e. pure MVS based on photoconsistency.

1. In chapter 4, I will present a novel multi-view depth estimation method that performs simultaneous depth estimation and denoising on narrow-baseline SE image sequences, i.e. sequences taken with a very fine angular resolution. This allows the method to deal with the often low local signal-to-noise ratio found in SEM images.
2. In chapter 5, I will show how a number of such depth maps can be used to reconstruct intricate and curved surfaces using a novel surface model based on local quadrics.
3. In chapter 6, I will present a novel surface reconstruction method that works on wide-baseline image grids taken from a range of rotation and tilt angles using both an SE and a BSE detector. The method focuses on fine surface features and curved surfaces.

The second part considers the shading found in SE and BSE images.

4. In chapter 7, I will present a novel empirical shading model for both SE and BSE reflectance and show how its parameters can be fitted to a sequence of images of a cylinder recorded at different rotation angles.
5. In chapter 8, I will show how my specific model formulation can be used to efficiently estimate the local surface normal from a set of observations.
6. In chapter 9, I will propose a depth estimation method that builds on *normal consistency* instead of photoconsistency, allowing it to estimate depths from images taken under a wide range of viewing angles.

Chapter 3

Projections and Conventions

In the following, I will briefly describe the notations used throughout this thesis. Much of the thesis deals with the mapping of 3D points to 2D images and vice-versa. There, I mostly follow the conventions that have been established by Hartley and Zisserman [75].

Points in 3D are denoted by capital letters and represented by column vectors, e.g. $X = (X_1, X_2, X_3)^t$. The corresponding coordinate system is referred to as *world space*. Points given in world coordinates are projected into *eye-space* coordinates $(e_1, e_2, e_3)^t$ through an affine transform

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = V \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}, \tag{3.1}$$

where V is a 3×4 *view matrix* specific to a given view. In eye space, the observer is located at the origin and looking at the scene in *positive* z-direction. The view matrix V is composed of an orthogonal 3×3 rotation matrix R and a column vector $t \in \mathbb{R}^3$ that represents a translation:

$$V = \begin{pmatrix} R & t \end{pmatrix}. \tag{3.2}$$

The origin of eye space is located at $-R^t t$ in world space. The eye space is primarily used when discussing surface normals. In those cases, I also use the symbol $\bar{V}$ to denote the normal matrix:

$$\bar{V} = \begin{pmatrix} & & & 0 \\ & R & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{3.3}$$

From eye space, a point is further projected into the *image space* coordinates $p = (x, y)^t$ via a projective transform:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{w} \begin{pmatrix} u \\ v \end{pmatrix}, \qquad \begin{pmatrix} u \\ v \\ w \end{pmatrix} = K \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = KV \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix}. \tag{3.4}$$

Here, K is an upper-triangular 3×3 camera matrix, and its last component $k_{3,3}$ is always equal to 1. Image space vectors are denoted by lowercase letters. The product KV is denoted by T, and the index i of the respective image is indicated in the subscript, e.g. T_i or V_i.

In certain places, a ray $r_{x,y,i}(z) : \mathbb{R} \mapsto \mathbb{R}^3$ is constructed that corresponds to a pixel (x, y) in a given image i and that maps different real-valued depths z to points in world space. It is given by

$$r_{x,y,i}(z) \coloneqq -R_i^t t + z R_i^t K_i^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \tag{3.5}$$

Lines l in image space are denoted by homogeneous row vectors. Then, a point x is located on the line when

$$lx = \begin{pmatrix} l_1 & l_2 & l_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0. \tag{3.6}$$

Planes in world space are denoted analogously.

Superscripts in parentheses denote additional qualifiers, and not exponents or derivatives. For example, $z^{(D)}$ denotes dense depth maps. Pixelwise access to images is denoted by subscripts – e.g. $z_p^{(D)}$ refers to the value of that depth map at pixel $p = (x_p, y_p)$. In a number of places, a 3D point is used as an argument to an image, e.g. $u_i(X)$. There, it means the interpolated value of image u_i at the 2D position to which point X projects in view i.

Chapter 4

Depth Estimation from Dense Image Sequences

The following chapter describes the shape reconstruction from geometrically calibrated narrow-baseline image sequences. A narrow baseline means that the angles between consecutive views are small. In my experiments, this angle was on average equal to 0.05°. In this scenario, we can safely assume that the gray value of a given point will only undergo a marginal change from frame to frame. The gray value of a point can therefore be used to identify that point in a sufficiently long subset of images around the reference image.

For that reason, the input sequence is subdivided into image batches, with the reference frame located in the middle. The depth map of each reference frame is computed exclusively from images within its batch. Choosing an overly long batch exposes us to the risk of excessive gray value changes, while a batch of insufficient length does not allow for a sufficiently precise triangulation of the depth. I have found batches comprising arcs of $\pm 1.25°$ around the reference frame to offer a reasonable trade-off between precision of triangulation and gray-value constancy.

Most traditional MVS methods estimate image correspondence by comparing image windows of a certain size. This makes the estimation robust to lighting changes, occlusion and noise, but it impairs the reconstruction of small features that do not cover an entire window. Methods that rely on individual pixels, on the other hand, generally require images with clearly distinguishable colors that remain constant over the entire sequence.

Since this thesis is concerned with the reconstruction of intricate shapes, the proposed methods should preferrably work on individual pixels. SEM images are always grayscale, and they contain a certain amount of noise. Although the most prominent features in those images are much stronger than the noise, many areas in the images show far fainter constrast.

In order to still be able to correctly match pixels in those areas, I have developed a depth estimation method that performs simultaneous depth estimation and denoising, producing both a depth estimate and a denoised gray value for each pixel. Both computations are performed simultaneously, so that my algorithm can use image structures for depth estimation that are fainter than the noise level of one individual image. Conversely, this also means that it can uncover hidden structures from a collection of images that are invisible in every single image alone.

The general framework for this type of depth estimation looks as follows. For each pixel xy in the reference image I_0, a ray through the scene can be defined that maps each depth

$z \in \mathbb{R}$ to a 3D point $p_{x,y,z} \in \mathbb{R}^3$. That point can be projected into every other image I_i and the encountered gray values can be extracted via cubic interpolation, yielding a set of values $v_{x,y,z,i}$.

Since the pixels are treated independently, the subscript xy can be dropped. Formally, the denoising depth estimation then consists of finding the maximum of the joint distribution $p(v, z|v_{z,i})$. The depth and denoised gray value are given by the depth z^* and value v^* that maximize this likelihood.

In the following section, I will first describe the statistical observation model that defines the relation between the observed gray values $v_{z,i}$ and the unknown true value v_0. This model contains an estimate of the noise amplitude as a function of the gray value, and it also accounts for outliers caused by occlusion.

The measurement of the noise parameters is described in section 4.2. Since that process already contains a depth estimation step using a simpler observation model, it also serves as a didactic example that should help the reader to better understand the method. The full observation model corresponds to a non-convex energy, so its global optimization is more involved than under the simplified model. Therefore, section 4.3 will describe an efficient algorithm to accomplish this.

The final result after this chapter will consist of a denoised image corresponding to the reference image, and a sparse depth map that describes the depth at each pixel where it can be known. If a pixel is located in a smooth area of the image, a certain range of depths around the true one all produce very similar gray values $v_{z,i}$. The true depth of those pixels therefore cannot be inferred from one pixel alone. The reconstruction of the entire surface from such sparse depth maps will be described in the next chapter.

4.1 Observation Model

Two distinct processes hamper the gray-value based identification of corresponding points: *noise* and *occlusions*. Noise refers specifically to pixelwise uncorrelated sensor noise, i.e. the fact that the gray value we observe is only a sample from a statistical distribution around the true value. The images are noisy because only a limited time (i.e. a few μs) is spent capturing electrons for each pixel, and only a finite number of electrons can be captured during that time. Because the process that determines whether a given electron is captured by a detector or whether it is missed is random, that number is associated with a specific uncertainty.

Occlusion refers to arbitrary surfaces from other parts of the object that can potentially be seen in front of the true point in any given view. If that is the case, then the observed value is completely independent of the true value.

Since the outcome of that random event is independent for each electron, this corresponds to a Poisson process, and would be optimally modelled by a Poisson distribution. Because of the very large number of electrons - and also in order to simplify the depth estimation algorithm - it is more practical to approximate the distribution by a normal distribution instead, with a standard deviation $\sigma(v_0)$ that is an affine function of the true value v_0. My model therefore assumes that the observed gray values are distributed according to a normal distribution around the true value, and that there is a certain minimal probability of observing *any* gray value.

Formally, let the distribution of observed gray values v be defined as follows:

$$p(v|v_0) = \frac{1}{N} \max\left(\theta, \exp(-\frac{|v - v_0|^2}{\sigma(v_0)^2})\right), \tag{4.1}$$

where v_0 is the true gray value, N is a normalization constant, θ is the outlier threshold and $\sigma(v)$ is the noise amplitude for gray value v. The noise amplitude is defined as

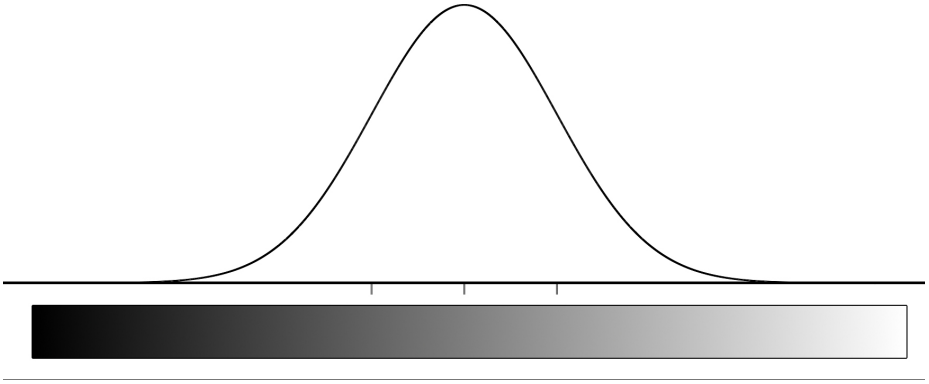
$$\sigma(v) = \sigma_0 + \sigma_S v, \tag{4.2}$$

where σ_0 and σ_S are measured from the input data. The area where $Np(v|v_0) > \theta(v_0)$ will be referred to as the *noise range* of v_0.

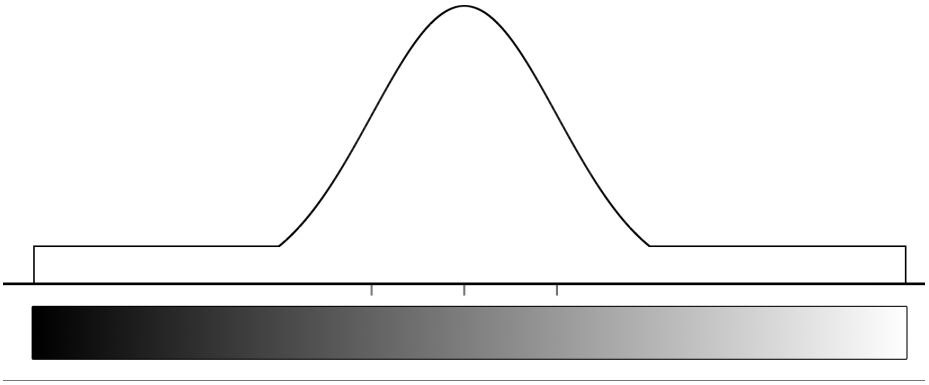
This distribution is an approximation of a mixture of a normal distribution representing noisy observations of the true surface and a uniform distribution representing occluded views. Although a sum of those two distributions would yield a more realistic model, i.e. the Blake-Zisserman function [76], the key advantage of the distribution in Eq. 4.1 is the fact that it is constant outside the noise range. This makes it more amenable to global optimization, as will be shown in 4.3.

We assume that the slope σ_S of the noise amplitude is sufficiently small compared to the noise range, so that $p(v|v_0)$ can be considered symmetrical, i.e. $p(v|v_0) \approx p(v_0|v)$. If the slope were zero, those two probabilities would indeed be identical. In reality, the small positive slope leads to a $p(v_0|v)$ that is slightly skewed to the right, but the assumption of symmetry is required for efficient optimization.

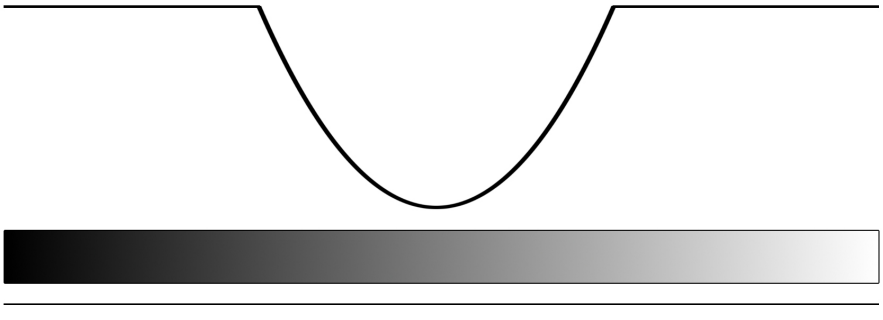
The outlier probability θ is fixed to e^{-4} for all images except for the reference. This guarantees that the Gaussian is cut off at two standard deviations. For the reference image, we keep θ zero, since a point can by definition never be occluded in the reference view; otherwise, that pixel would refer to the occluding point in front of it. To account for this distinction, I will refer to θ as θ_i going forward.



Non-robust Gaussian distribution $p_S(v|v_0)$.



Robust distribution $p(v|v_0)$.



Robust energy $E(v|v_0)$.

Figure 4.1: Illustrations of the functions described in the text.

4.2 Estimation of Noise Intensity

In the following section, I will describe how the noise parameters σ_0 and σ_S are measured from a short sequence of geometrically calibrated input images I_i. A secondary purpose of this section is to serve as a simplified version of the actual depth estimation method that will be presented in the next chapter.

The noise estimation algorithm essentially consists of a depth estimation procedure where the observation model $p(v|v_0)$ has been replaced by a simple normal distribution of constant σ, followed by a measurement of the noise using the image correspondence implied by those depths. Note that the value of the constant σ is not relevant, as it does not influence the optimum in Eq. 4.4. It has therefore been set to 1 for the sake of simplicity.

4.2.1 Simplified Depth Estimation

As mentioned at the beginning of the chapter, the depth estimation consists of finding the maximum of the joint distribution $p(v, z)$ of the gray value v and depth z for every pixel. We assume a normal distribution as our simplified observation model,

$$p_S(v|v_0) = \frac{1}{2\pi} \exp(-(v - v_0)^2). \tag{4.3}$$

This is equivalent to Eq. 4.1 with the outlier threshold θ set to 0 and the variance σ set to 1.

Under this assumption, the computation of the maximum along the v direction for a given depth z becomes trivial. Every such depth corresponds to a set of gray values $v_{z,i}$. The most likely true value v_z^* is the most likely v_0 to give rise to those values. Since we have assumed the measurements to be uncorrelated, we can write,

$$v_z^* = \underset{v_0}{\operatorname{argmax}} \prod_i p_S(v_{z,i}|v_0) = \underset{v_0}{\operatorname{argmin}} \sum_i (v_{z,i} - v_0)^2 = \mu_z. \tag{4.4}$$

Under the simplified assumption, v_z^* is equal to the mean μ_z of the observed values $v_{z,i}$. The most likely depth z^* is then given by the depth of the most likely μ_z, which is the depth of least variance,

$$z^* = \underset{z}{\operatorname{argmin}}\, \sigma_z^2, \tag{4.5}$$

$$\sigma_z^2 = \frac{1}{\#i - 1} \sum_i (v_{z,i} - \mu_z)^2. \tag{4.6}$$

This depth is only reliable if the pixel x lies in an area with sufficient image contrast. Otherwise, multiple depths around z^* produce very similar gray values, and thus very similar means and variances. This is only a problem if the actual depth is of interest, however. For the purpose of noise estimation, the unreliable depth is of no concern, as long as the $v_{z,i}$ do not deviate excessively from the true ones.

4.2.2 Noise Estimation from Depths

If the surface point behind pixel x is not visible throughout the entire sequence, then its minimal variance σ^{2*} will still be comparatively large. For that reason, only pixels with a variance beneath a certain threshold τ are considered from this point forward. This filtering step removes all observations of that point, not only those in the occluded views. Hence, an occlusion only has to be certain in some of the views for that point to be removed completely.

Since we are working with a short narrow-baseline sequence, we can assume that many pixels will still remain valid, in spite of the aggressive filtering step. At the same time, incorrect pixels that have survived that filtering step have done so because of their low variance. As a consequence, the extent to which they can distort the final estimate is limited.

Next, all the pixels x are collected into bins b based on their mean value, μ^*. Each bin b corresponds to a bin gray value, v_b. For each bin, a new variance σ_b^2 is computed from the differences between the values $v_{z^*,i}$ observed at the optimal depth z^* and their respective means μ^*. Due to the discretization of the bins, that mean is not exactly equal to the bin gray value.

Finally, the intercept σ_0 and slope σ_S are estimated through weighted linear regression. This is done by finding a σ_0 and σ_S that minimize

$$\sum_b w_b(\sigma_0 + \sigma_S v_b - \sigma_b)^2, \tag{4.7}$$

where w_b is the number of pixels that have contributed to bin b.

4.3 Occlusion-Robust Depth Estimation

The denoised values v^* that were estimated under the simplified model are mere averages of all observed values $v_{z^*,i}$. If any of those values is produced by an occluding surface, then the denoised value will be contaminated by the gray value of the occluder. In that case, v^* will strike a compromise between the true value and the outlier. As a consequence, the variance will be formed around the wrong mean, and the depth of least variance will therefore also be unreliable.

Under the full observation model, estimating the most likely gray value v^* is no longer trivial. The observations are still uncorrelated, so we can still write,

$$v_z^* = \operatorname*{argmax}_{v_0} \prod_i p(v_{z,i}|v_0) = \operatorname*{argmin}_{v_0} \sum_i E(v_{z,i}|v_0) \tag{4.8}$$

The energy E is given by,

$$E(v_{z,i}|v_0) = -\log(p(v_{z,i}|v_0)) = \min\Big(-\log(\theta_i), \frac{(v_{z,i}-v_0)^2}{\sigma(v_0)^2}\Big) - \log(N), \tag{4.9}$$

and it takes the form of a truncated parabola. Under the simplified model, the sum of the (not truncated) parabolae was still a parabola and its minimum was located at the mean of the $v_{z,i}$. Under the full model, there is no longer an analytical solution for the minimum v_z^*.

The energy E is also not a convex function, so a local optimization would not converge from all initial positions. In order to avoid a costly exhaustive search that would have to be repeated for each depth sample of each of the millions of pixels, we need to slightly alter the definition of E.

4.3.1 Shifted Energy

Since we are only interested in the minimum of the sum of the individual contributions of E, we can shift each energy where $\theta_i > 0$ by a constant offset,

$$E_Z(v_0|v) \coloneqq E(v_0|v) + \log(N\theta_i) \tag{4.10}$$

$$= \min\left(0, \frac{(v - v_0)^2}{\sigma(v)^2} + \log(N\theta_i)\right). \tag{4.11}$$

The shifted energy E_Z is now zero outside the noise range of its associated observation v and negative inside. This allows for an efficient non-iterative algorithm. This shift does not influence the position of the optimum.

4.3.2 Optimization

Using the shifted energy E_Z, the most likely true gray value v^* to have produced a set of observations $v_{z,i}$ can be computed efficiently through Eq. 4.8.

To this end, we discretize the value range of v and we allocate one scalar energy value e_v for each value sample. The denoising behavior of my algorithm allows it to determine values that are more precise than the signal resolution of a single source image. For that reason, even though I am working with 8-bit images, more than 256 samples would be needed to cover the entire value range.

Since a point can by definition never be occluded in the reference view, we know that the true value must be close to the value $v_{z,0}$ seen there, so that only values close to it need to be considered. In my implementation, I have chosen the range to be centered at $v_{z,0}$, i.e. the value observed in the reference view, and six standard deviations $\sigma(v_{z,0})$ wide.

The energies e_v are initialized by $E(v_0, v_{z,0})$. As the point can never be occluded in the reference view, θ_0 is equal to zero there, so the unshifted energy E has to be used in that case.

Next, the algorithm iterates over all observations $v_{z,i}$, and we determine the noise range $[v_{\min}, v_{\max}]$ of each observation. For every value sample within that interval, its energy contribution $E_Z(v_0, v_{z,i})$ is added to the e_v. Since the shifted energy E_Z is zero outside of the interval, we know that the observation cannot contribute to any e_v there. Hence, my algorithm only needs to update the e_v of value samples within the noise range.

The outlier threshold θ_i has been fixed to two standard deviations, so $v_{\min}$ and $v_{\max}$ are given by

$$v_{\min}, v_{\max} = v_{z,i} \pm 2\sigma(v_{z,i}). \tag{4.12}$$

After all the observations have been considered, the smallest e_v is chosen as the energy e_z of depth z, and its corresponding gray value v is chosen as the value v_z at that depth.

This procedure is repeated for all depth samples z, yielding a sequence of depth energies e_z and depth gray values v_z. The depth corresponding to the smallest energy e_z is then chosen as the discrete depth estimate z_D^*, and its gray value is chosen as the denoised pixel value v^*.

Finally, as is common in depth estimation algorithms, a more precise depth estimate z^* is computed through quadratic interpolation. This is done by finding the vertex of a 1D

parabola that passes through the energy values $e_{z_D^*}$ and its two immediate neighbors, $e_{(z_D^*-\delta z)}$ and $e_{(z_D^*+\delta z)}$:

$$z^* \coloneqq z_D^* + \frac{\delta z}{2} \frac{e_{(z_D^*-\delta z)} - e_{(z_D^*+\delta z)}}{e_{(z_D^*-\delta z)} + e_{(z_D^*+\delta z)} - 2e_{z_D^*}}, \tag{4.13}$$

where δz is the step size in z direction.

4.3.3 Confidence Estimation

Although both the depth and the denoised gray value are now known for each pixel, the algorithms that will be presented in the following chapter also require a measure of confidence. The confidence c of the depth estimate is determined as follows.

The depth energies e_z are converted into probabilites $p_z \coloneqq \exp(-\beta(e_z - e_z^*))$ which are then normalized along the z direction. Here, β is an additional sharpening parameter, inspired by the thermodynamic coldness β in thermodynamics. The normalization corresponds to the assumption that one of the depth values has to be the true one.

From those probabilites, the variance σ_z^2 around z^* is computed,

$$\sigma_z^2 \coloneqq \frac{1}{1 - \sum\limits_z p_z^2} \sum_z p_z (z - z^*)^2, \tag{4.14}$$

And the confidence c is then given by

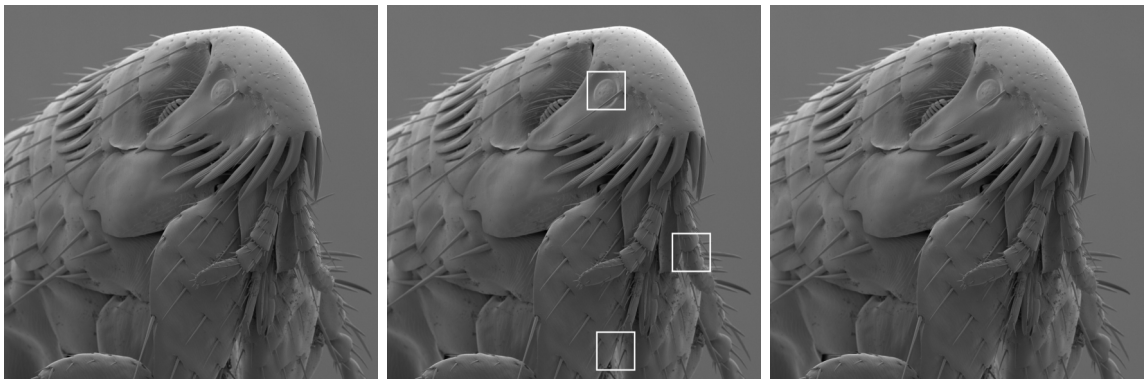
$$c_p \coloneqq \begin{cases} \sigma_z^{-2} & \text{if } \sigma_z^2 < \theta_\sigma \\ 0 & \text{otherwise.} \end{cases} \tag{4.15}$$

The computation of the variance corresponds to fitting a normal distribution to p_z under a known mean, while the thresholding by θ_σ can be understood as a test on whether the normal distribution is an appropriate approximation of p_z.

4.4 Experiments

A sequence showing a cat flea was recorded, consisting of 1400 SE images spaced at 0.05° angular intervals. The capture of these images was performed by Martin Oeggerli and Ken Goldie. I have calibrated the sequence geometrically by manually selecting a number of corresponding points in every 100^{th} frame, and then performing a bundle adjustment procedure on this sparse set. The calibration of the intermediate frames was obtained through a Farnebäck dense optical flow [77] computed between neighboring frames in both directions. Only pixels with a sufficiently small roundtrip error after being mapped through the 100 frame sub-sequence and back were considered for the calibration of the intermediate frames. No lens distortion model was applied.

The sequence was subdivided into 108 overlapping batches of 51 images. The center image of each batch was selected as the reference. The proposed algorithm was applied to each batch, resulting in a sparse depth map and a denoised version of the image. A number of results are shown below. The computation of a single depth map took approximately 14 hours using the robust model and slightly more than 3.5 hours with the simplified model on a single CPU of an Intel Xeon E5-1620 running at 3.6GHz.



first frame reference frame last frame

Figure 4.2: Three out of 51 input images that constitute a batch. The squares correspond to the closeup shown in Figs. 4.6 and 4.7. Readers who are able to look at neighboring pairs of images cross-eyed will observe a 3D effect.

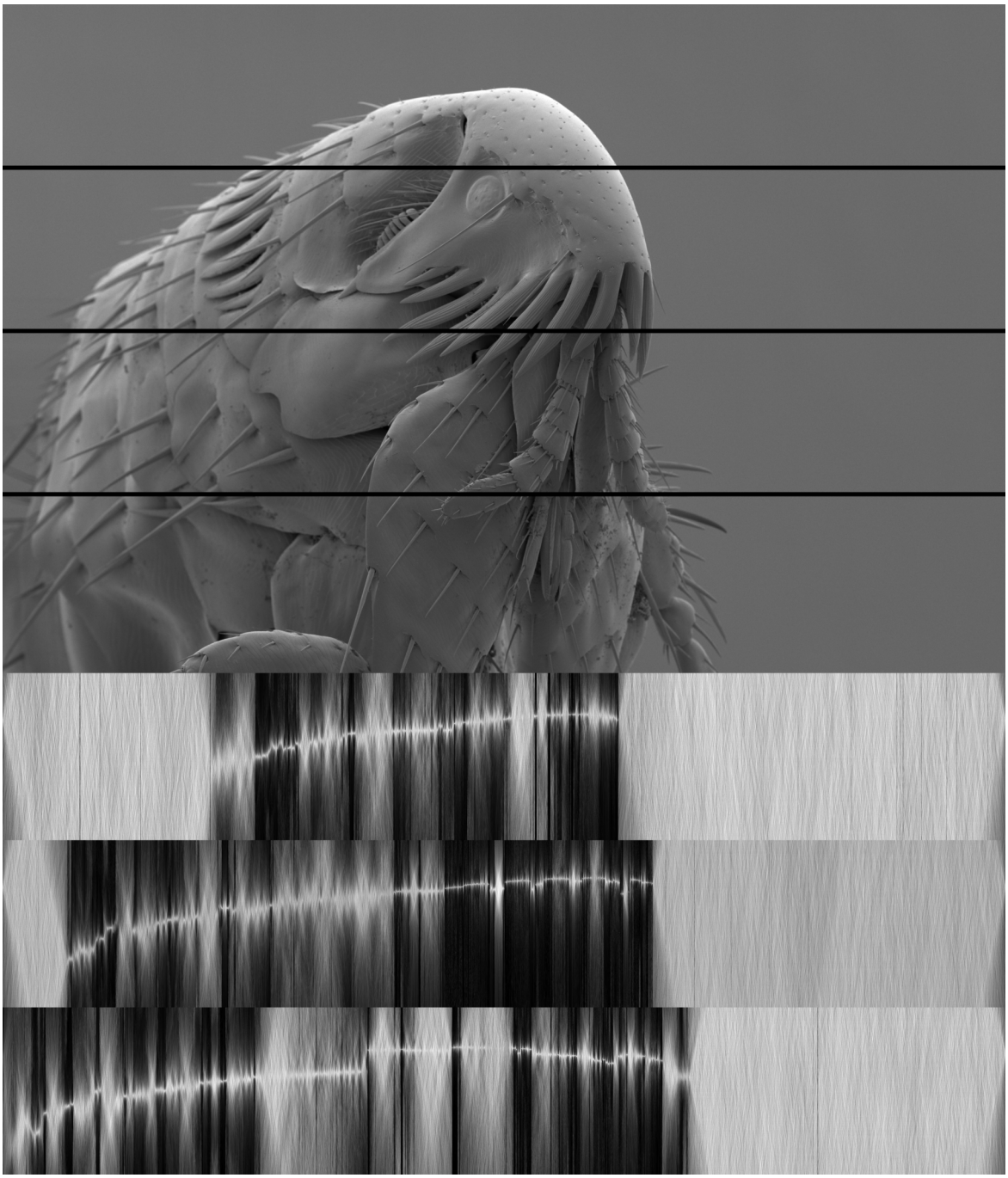


Figure 4.3: Slices through the robust energy $e_{x,z}$ at the y values indicated by the black lines. The x-axes of the images and the diagrams coincide, while points lower in the diagram are further away. The vertical scale of the diagrams is arbitrary but equal. Darker points indicate higher energies.

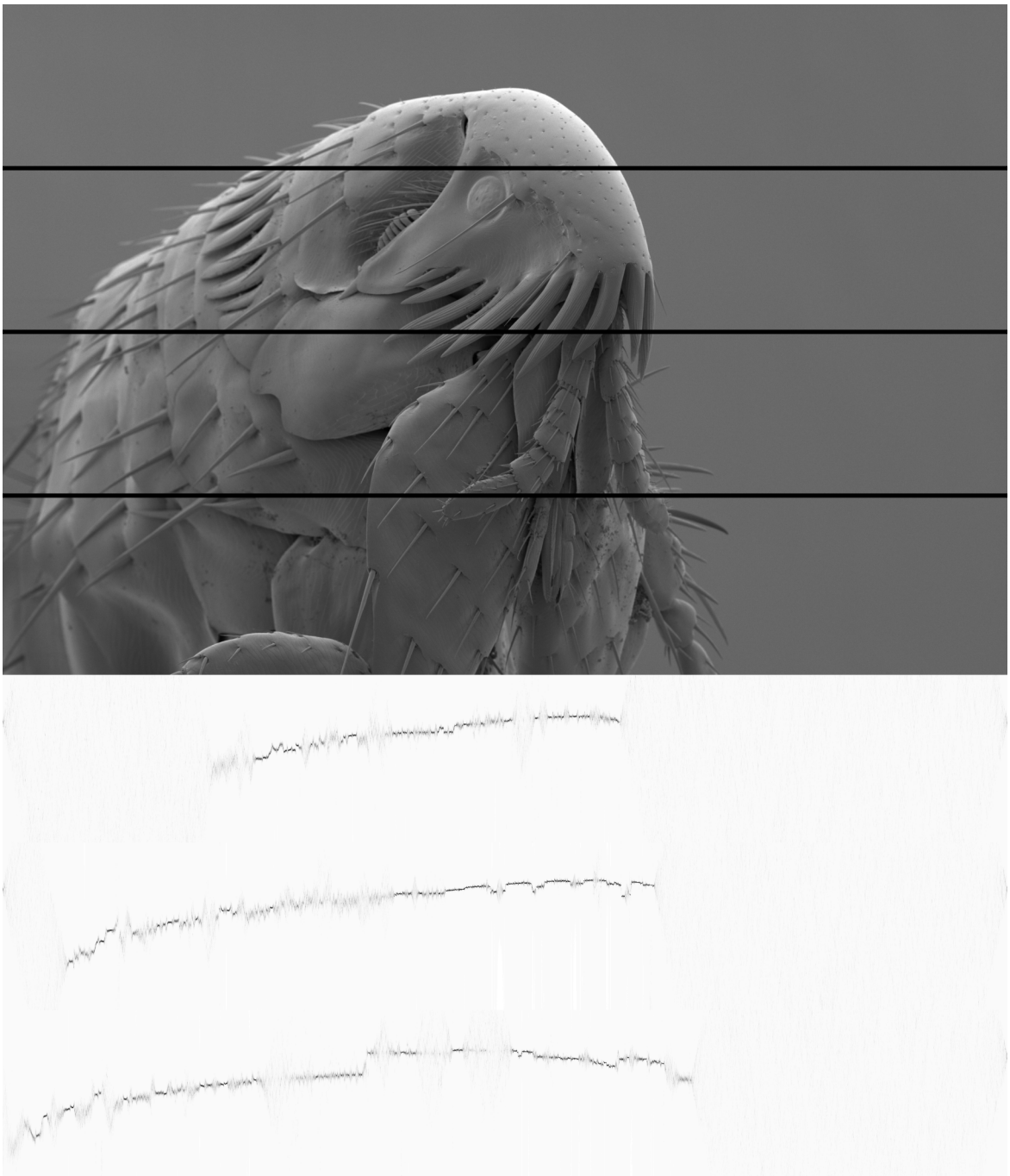


Figure 4.4: Slices through the probabilites $p_{x,z}$ corresponding to the energies in Fig. 4.3. Darker points indicate a greater probability. Note that reliable depth estimates are missing in the smooth areas of the image.

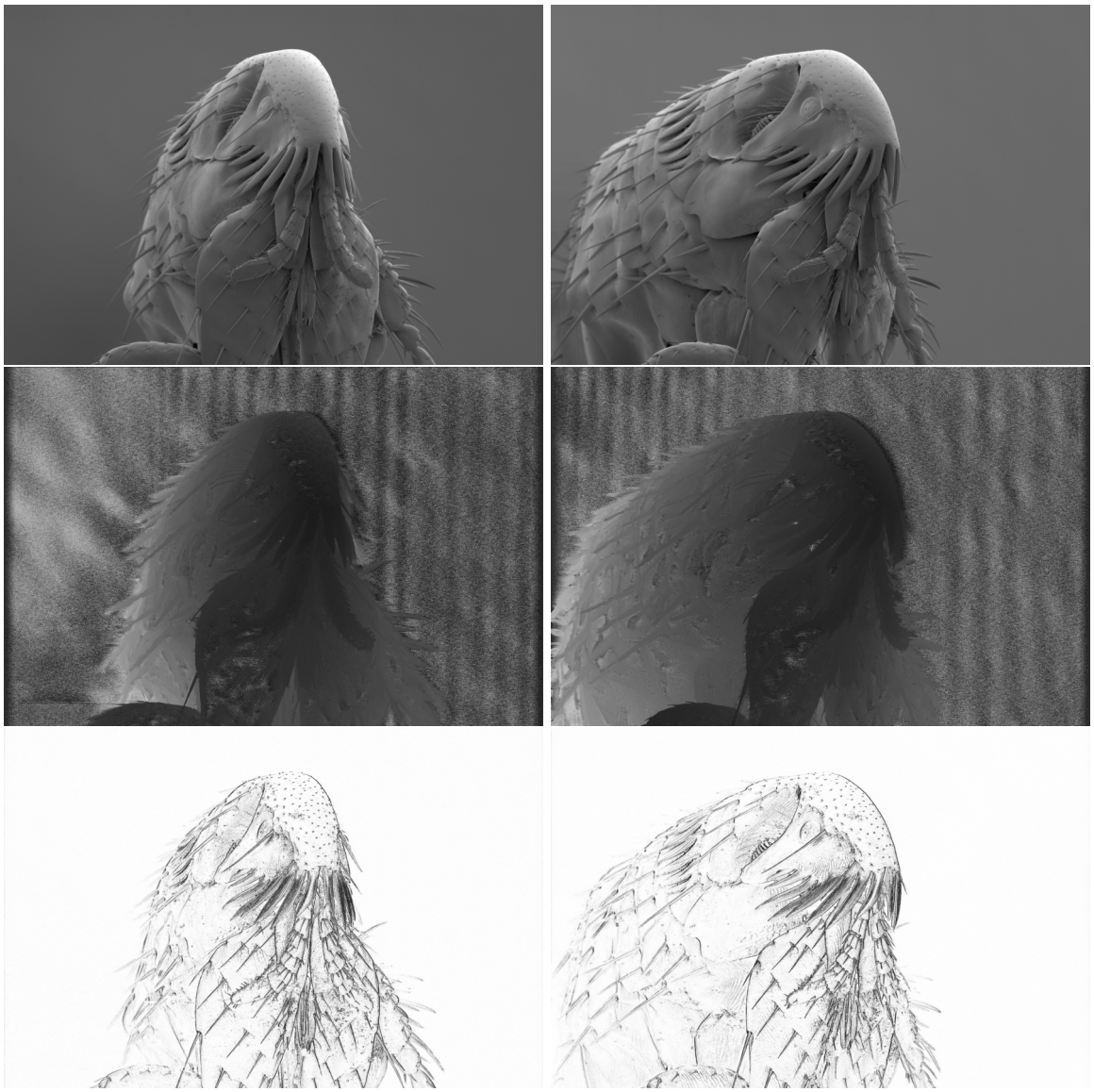
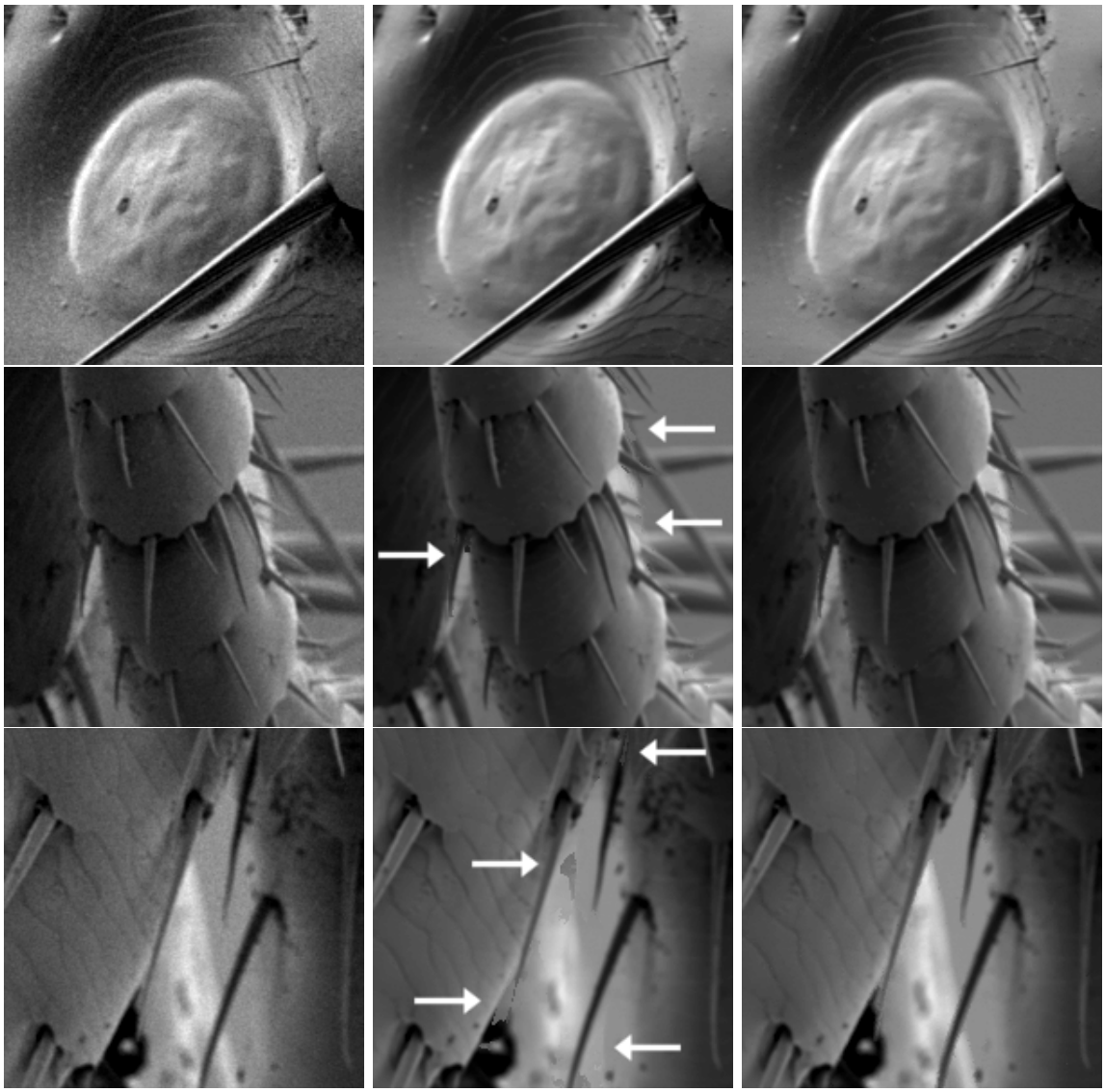


Figure 4.5: Reconstruction of two batches. The reference frame (top), the depth map (center) and its confidence (bottom). Darker pixels indicate closer points in the depth map and a higher confidence in the confidence map. Note that the noise in the depth map is contained to areas that are smooth in the reference frame, where the confidence is also low.



original simplified robust

Figure 4.6: Denoising results. Original image (left), image denoised using the simplified model (center) and the robust model (right). Note the artifacts in the presence of occlusion. The contrast has been increased to showcase the effects of denoising.

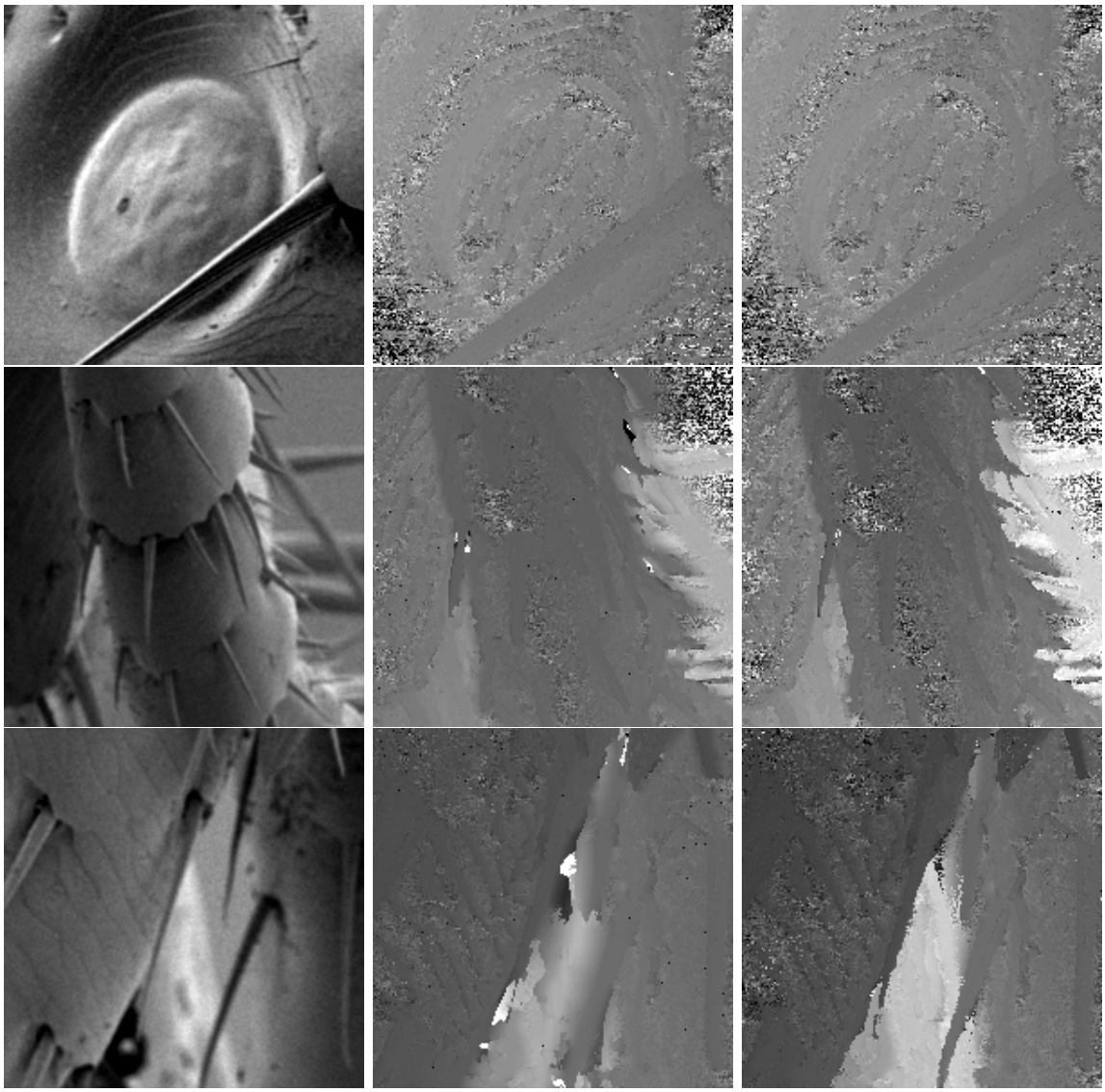


image simplified robust

Figure 4.7: The estimated depth. Input image (left), depth obtained under the simplified model (center) and the robust model (right).

4.5 Conclusions

In this chapter, I have shown a depth estimation method that performs simultaneous depth estimation and denoising using an occlusion-robust and a simplified observation model. Although the robust estimation is four times more expensive, it is able to obtain reliable depth estimates and denoised values in areas that are not visible throughout the entire batch. The artifacts of the simplified model are more noticeable in the depth maps than in the denoised images. Under both models, the depth is only reliable near edges and it takes on nearly random values in large smooth areas. I will therefore refer to these depth maps as *sparse depth maps* from now on.

In the next chapter, I will describe how the sparse depth maps can be interpolated along the structure of the denoised images, resulting in truly dense depth maps. Those will then be utilized for a volumetric surface reconstruction procedure.

Chapter 5

Shape Reconstruction From Dense Sequences

In the following chapter, I will describe a method that aims to reconstruct watertight surfaces from the depth maps estimated in the previous chapter. In particular, this method is directed at **curved geometry** featuring **thin protrusions** and **sparsely textured surfaces**. The latter property means that large surface areas appear very smooth in the images, so that their depths can only be determined at their edges. In SEM images, such areas often appear at depths far away from the focal plane. There, only high-contrast edges can be observed while all fainter edges are blurred beyond perception.

The method also takes into account that only a **narrow range of input views** are available. This is a neccessary consequence of the high angular resolution of the image sequences and any given time budget. The capture of the 1400 images used in my experiments took approximately three days on an FEI Versa SEM, and it shows a 70°-arc sampled in 0.05° intervals. If the same resolution were to be obtained along a 2D grid of angles, i.e. $70° \times 70°$, then the capture of the corresponding $1.96 \cdot 10^6$ images would take more than ten years.

In order to deal with curved surfaces seen from a narrow range of angles, my method uses quadrics to model curved contours. This allows it to extrapolate the curvature implied by multiple corresponding contour observations beyond the angles under which the contours can be observed. This quadric-based approach can be understood as a form of third-order regularization. The classical Poisson reconstruction approach corresponds to first-order regularization, since it entails finding a scalar field that exhibits a small gradient length everywhere. A second-order approach would consist in finding a scalar field with a small Frobenius norm of the Hessian matrix. Such a method has been proposed by Schrörs et al. [78], and it has been shown to reconstruct smoothly curved surfaces well. Quadrics are defined as isosurfaces of quadratic 3D functions which exibit a constant Hessian. By looking for local quadrics in space, I am in essence minimizing local changes in the Hessian matrix itself, while allowing for arbitrarily large entries in it. Unlike the methods mentioned above, this is not done through a variational approach, but through a conceptually simpler sliding-window approach.

My method comprises three individual steps, outlined as follows:

1. **Depth map interpolation**: for each input view, a novel depth map is computed that is consistent with the per-pixel depth values at high-confidence pixels and whose depth contours coincide with the edges observed in the denoised images.

2. **Local quadric estimation**: quadrics are estimated that conform locally to the contours seen in the interpolated depth maps. These quadrics are used to model surface areas of high curvature, i.e. thin protrusions.

3. **Volumetric reconstruction**: the quadrics are combined with the depth information from the depth maps to obtain a final watertight surface. Areas of high curvature are reconstructed from the quadrics, while the smoother areas are reconstructed using information from the interpolated depth maps.

These steps will be elucidated and further motivated in the following.

5.1 Depth Map Interpolation

The depth maps that have been estimated in the previous chapter are only reliable for pixels that show a certain amount of image contrast. Although a depth is technically known for each pixel, I will refer to those depths z^* as **sparse depth maps** from now on. The new depth maps that are computed by the method presented in this section will be referred to as **dense depth maps**.

Because the depths are known for pixels where the gray value intensity exhibits a significant gradient magnitude, they are known at the edges seen in the image, but not in the smooth areas. More precisely, only edges that are *not parallel* to the epipolar direction, i.e. the direction of perceived image motion, count as edges in this context. In my experiments, the object is rotating around a vertical axis, so the epipolar direction is always horizontal. Horizontal edges thus do not provide any information about their depth.

The aim of the depth interpolation algorithm is to propagate this depth information known at the edges into the smooth areas that separate them. This is accomplished by finding a new 2D scalar field $z_p^{(D)}$ that assumes values similar to those of z_p^* for pixels $p = (x, y)$ where the confidence c_p is high, and that exhibits a similar value for neighboring pixels p and q unless the corresponding denoised values v_p^* and v_q^* also differ significantly.

Formally, this corresponds to the minimization of the following discrete energy,

$$E_{\text{fill}}(z^{(D)}) \coloneqq \sum_p \Big(c_p (z_p^{(D)} - z_p^*)^2 + \mu_D \sum_{q \in N(p)} a_{pq} (z_p^{(D)} - z_q^{(D)})^2 \Big), \tag{5.1}$$

where μ_D is a regularization scale and $N(p)$ is the Moore neighborhood of pixel p, i.e. its eight immediate neighbors. The pairwise pixel-affinities a_{pq} are given by

$$a_{pq} \coloneqq \exp(-\frac{1}{\lambda^2}(v_p^* - v_q^*)^2), \tag{5.2}$$

where λ is a contrast parameter that describes the sensitivity of the regularizer to the image structure. Because the image v^* has been denoised, a very low value can be used here.

Unlike the similar depth interpolation step proposed by Shan et al. as part of their extension to Poisson reconstruction [20], my formulation is discrete and, when viewed as a discretization of a continuous energy, it corresponds to a penalty on the *first* derivative of $z^{(D)}$. Their energy is formulated in a continuous way and it penalizes the diagonal terms of the *second* derivative. My formulation can thus handle features that are only one pixel in size, while theirs requires an area of sufficient size to compute a Hessian matrix.

I do, however, apply the sky term proposed in that work. This means that a set of pixels that certainly belong to the background (i.e. the sky in their work) are marked manually and set to a very large depth value and a high confidence. This ensures that the uniform image areas around the object are reconstructed as the background. Without that term, there is usually not enough contrast in the background to estimate its depth. In practice, those areas are only marked once and the same mask is applied for all images.

This sky term produces very large depth differences across the outer contours of the object. This can have a destructive effect on the resulting depth maps, since it tends to pull foreground features down towards the background. At the same time, it makes small protruding features stand out better against the background, so I have chosen to use it anyway and rely on the large number of depth maps to alleviate its destructive effects.

The energy E_{fill} is convex and it is minimized using an iterative Jacobi method, i.e. the value $z_p^{(D)}$ of each pixel is iteratively set to the optimum given the current values of its eight neighbors,

$$z_p^{(D)}[t+1] \coloneqq \frac{c_p z_p^*[t] + \mu_D \sum\limits_{q \in N(p)} a_{p,q} z_q^{(D)}[t]}{c_p + \mu_D \sum\limits_{q \in N(p)} a_{p,q}}. \tag{5.3}$$

The procedure is aborted once the smallest value change $\min_p |z_p^{(D)}[t+1] - z_p^{(D)}[t]|$ in an iteration falls beneath a certain threshold $\delta_{\min}$. This optimization process is accelerated through a hierarchical multigrid approach, i.e. the optimal $z^{(D)}$ is first determined at a lower resolution, that solution is upsampled to the next resolution level and it is then used as the initial condition for the optimization at that resolution level.

In addition to the dense depth maps, I also interpolate the sparse confidence c itself by minimizing,

$$E_{\text{fill}}(c^{(D)}) \coloneqq \sum_p \Big(c_p (c_p^{(D)} - c_p)^2 + \mu_D \sum_{q \in N(p)} a_{pq} (c_p^{(D)} - c_q^{(D)})^2 \Big). \tag{5.4}$$

The interpolated confidence $c^{(D)}$ then indicates which points have received high-confidence information from their neighbors and is used in the surface reconstruction process in the end of this chapter.

The resulting dense depth maps $z^{(D)}$ are smooth in low-confidence areas, and their contours coincide with the edges in v^*. The depths also carry a fronto-planarity bias, which leads to a positive depth bias in convex curved areas and to a negative bias in concave areas. In the extreme case of smooth, untextured cylindrical features, this results in flat ribbons that are located at the depth at which the contour of the cylinder is seen.

Furthermore, thin protruding features that do not exhibit sufficient contrast are shifted backwards. This has three reasons. First, the low contrast does not allow for a sufficient confidence. Second, their small image area (i.e. number of pixels) does not lead to a sufficient increase in energy when they are dislocated. Third, the gray value difference does not reduce the pairwise affinity a sufficiently if the contrast is too low.

Because of these limitations, the dense depth maps alone do not provide a sufficient description of the object. They are still necessary for the following two purposes.

First, they tell us the depth of sparsely textured areas. Without this depth interpolation, I have found that the final watertight surface will often collapse to a skeletonized version, i.e. one that shows material underneath the visible edges, while the smooth areas, as well as the

majority of the enclosed volume, is wrongfully classified as empty space. In the absence of dense depth maps, this version is an equally plausible interpretation of the sparse depths as the correct one.

Second, contours can be reliably detected in the dense depth maps and their lateral (i.e. image-space) locations can be extracted from them. Such contours can then be used to fit local quadrics, as will be shown in the following.

5.2 Local Quadric Estimation

5.2.1 Contour Detection

The 3D curve that we perceive as the contour of a curved surface slides across the surface as the point of view changes. If the image stream offers sufficient angular resolution, then the depth of such a curve can still be determined. I will refer to those depths as *transient depths*, since they are only valid for one moment in time.

In the following, I will describe how to detect a set of image points representing likely contour candidates and how to estimate their transient depth. Later, these contour candidates will be used to fit local quadrics.

To detect the contours, I apply the non-maximum suppression step defined by the Canny edge detector [79] to the dense depth map $z^{(D)}$. This provides a set of potential contour pixels $p_c = (x_c, y_c)$. For every such contour candidate, its transient depth is estimated from the initial sparse depth samples z^*, and not from the dense depth map $z^{(D)}$. This helps to preserve small features that are dislocated towards the background during depth map interpolation.

The transient depth $z_p^{(T)}$ of every contour pixel p is estimated using a sliding window approach,

$$z_p^{(T)} \coloneqq \frac{1}{w_p} \sum_{q \in W(p)} w_{pq} z_q^*, \tag{5.5}$$

$$w_p \coloneqq \sum_{q \in W(p)} w_{pq} \tag{5.6}$$

where $W(p)$ represents a window of width $4\sigma_c$ centered at pixel p and the pixel weights w_{pq} are given by

$$w_{pq} \coloneqq c_q \gamma_q \exp\Big(- \frac{|q-p|^2}{\sigma_c^2} \Big) \tag{5.7}$$

$$\gamma_q \coloneqq \max\Big(0, (G_{\sigma_c} * z^*)_q - z_q^*\Big). \tag{5.8}$$

Here, the distance $|q-p|$ is measured in image space, and $G_{\sigma_c}*$ represents a convolution with a Gaussian. The clamped difference of Gaussians γ serves to select foreground pixels, i.e. pixels that are closer to the observer than the average in their area.

The edge candidates are then filtered by computing the total confidence,

$$c_p^{(C)} \coloneqq |\nabla z_p^*| \Big(1 - \exp(-\frac{w_p^2}{\sigma_w^2})\Big) \exp\Big(- \frac{\nu_p^2}{\sigma_\nu^2}\Big), \tag{5.9}$$

where σ_w^2 and σ_ν^2 are two parameters that control the sensitivity to the two criteria, and ν^2 is the w_{pq}-weighted variance of z^* around $z^{(T)}$ which is given by

$$\nu_p^2 \coloneqq \frac{w_p}{w_p^2 - \sum\limits_{q \in W(p)} w_{pq}^2} \sum_{q \in W(p)} w_{pq} (z_q^* - z_p^{(T)})^2. \tag{5.10}$$

This excludes candidates that are either insufficiently supported by evidence (w is small) or that contradict too much evidence (ν^2 is large).

For each contour candidate $p_c = (x_c, y_c)$, the corresponding 3D normal n is computed from the image-space gradient of z^*, $g = (g_x, g_y)^t \coloneqq \nabla z^*$, under the assumption that n_c is perpendicular to the viewing direction. This is always the case if p_c represents a contour.

To that end, we define an image-space line l in homogeneous coordinates by $lp_H = 0$, $p_H = (x, y, 1)^t$ that is perpendicular to g and that passes through p_c. That line is given by the row vector $(-g_x, -g_y, g_x x_c + g_y y_c)$. It corresponds to a plane in space $\pi = lT_i$. The surface normal n_c is then given by the normal of that plane, i.e. the first three components of π.

Together with X_c, the back-projected position of p_c, the normal represents an oriented surface point that will be used to estimate local quadrics, as will be shown in the following.

5.2.2 Voxelwise Quadrics

The operations so far have taken place on individual depth maps. Each of them corresponds to one of the short image batches that were defined at the beginning of chapter 4. From this point forward, the information from all those depth maps is merged together to obtain a reconstruction of the entire surface.

The next operation aims to estimate local quadrics from the contour candidates of all images. This is done in the voxels of a discretized volume. An arbitrary quadric can be written as the level set $f(X) = X^t C X = 0$, where $X = (x_1, x_2, x_3, 1)^t$ is a homogeneous position vector and C is a real, symmetrical 4×4 matrix that holds the coefficients of the quadric.

We are now looking for a matrix-valued field $C(x, y, z)$ that varies smoothly across space and corresponds to quadrics consistent with the detected contour candidates. This is done by estimating a quadric C for each voxel, relying on information provided by nearby edge candidates. This corresponds to another sliding-window fit.

I have found that more reliable results can be obtained by estimating the gradient of $f(X)$ first, and then the integration constant. Estimation of the full quadric at once tends to produce unwanted folding-over effects if the contours are misaligned. The gradient of $f(X)$ is parallel to the surface normal of the quadric and is given by the first three components of $2CX$, to which I will refer as $(2CX)_{xyz}$.

To estimate the gradient of the quadric, I minimize the following expression over the contours in a neighborhood N of each grid point X_G,

$$\sum_{X_c \in N(X_G)} w_c ((2CX_c)_{xyz} - n_c)^2, \qquad w_c = c_c \exp(-\frac{1}{\sigma_q^2} |X_c - X_G|^2). \tag{5.11}$$

This yields the gradient of a quadric that conforms to the contour normals n_c at the locations of the contours X_c, weighted by their confidence c_c and their distance to X_G. This

computation is performed in closed form. In addition to C_{xyz}, I also store the sum of all contour weights w_c, to which I will refer as w_q, the weight of the quadric.

Since C is symmetrical, we are now only missing one last coefficient, $c_{4,4}$. It is computed as the median of $-X_c^t C_0 X_c$ over all contours X_c in $N(X_G)$; C_0 is equal to C with its unknown last coefficient set to zero. The quadric matrix estimated at grid point X_G will be referred to as C_G.

5.3 Watertight Surface

Finally, my algorithm estimates a watertight surface that conforms to the quadrics in the areas surrounding the detected contours and to the dense depth maps elsewhere. This is done by solving a Poisson problem using specific regional terms. An optimal 3D scalar field f is found by minimizing

$$\int_V \mu_q c_q(X)(f(X) - q(X))^2 + \mu_\phi c_\phi(X)(f(X) - \phi(X))^2 + \mu_r |\nabla f(X)|^2 dX, \tag{5.12}$$

where V is the volume covered by the voxel grid, $q(X)$ and $\phi(X)$ are the quadric regional term and the flat regional term, $c_q(X)$ and $c_\phi(X)$ are their confidences, and μ_q and μ_ϕ are the corresponding global weights, while μ_r controls the amount of regularization.

The quadric regional term $q(X)$ and its confidence $c_q(X)$ are computed using the following quadric voting scheme:

$$\psi(x) = \sum_{X_G \in N(X)} w_q \exp(-\frac{1}{\sigma_q^2}|X - X_G|^2) \operatorname{sign}(X^T C_G X), \tag{5.13}$$

$$q(X) = \operatorname{sign}(\psi(X)), \tag{5.14}$$

$$c_q(X) = |\psi(X)|. \tag{5.15}$$

This can be understood as follows. For every point X, every surrounding grid point X_G casts a vote on whether X belongs inside or outside the quadric C_G. Those votes are weighted based on the distance between X and X_G and proportionally to the weight of the quadric, w_q.

The flat regional term $\phi(X)$ is computed from the dense depth maps $z^{(D)}$ and the filled confidence maps $c^{(D)}$. For every depth map, a flatness term $h(x, y)$ is computed from the contour confidence c_c as

$$h(x, y) = \exp(-(G_{\sigma_h} * c_c)^2(x, y)/\eta_h^2), \tag{5.16}$$

where $G_{\sigma_h}*$ denotes a convolution with a Gaussian and the threshold η_h is set low enough to exclude all areas that contain a significant contour confidence c_c.

For every 3D point X and image i, let $\delta_i(X)$ be the depth difference between X and the depth map, with a positive sign if X is located behind the surface. We define the flat regional term and its confidence as follows:

$$\chi(x) = \sum_i \begin{cases} \zeta h_i(X) c_i^{(D)}(X) \exp(-\delta_i(X)^2/\tau^2), & \text{if } \delta_i(X) > 0 \\ h_i(X) c_i^{(D)}(X)(1 - \exp(-\delta_i(X)^2/\rho^2)), & \text{otherwise} \end{cases} \tag{5.17}$$

$$\phi(X) = \operatorname{sign}(\chi(X)), \tag{5.18}$$

$$c_\phi(X) = |\chi(X)|. \tag{5.19}$$

If X is located behind the depth map, then we assume it to be inside the material with a certainty that decays as the distance increases. If X is located in front of the depth map, then the certainty of being outside increases with the distance. The parameter τ describes the expected thickness of the material, while ρ describes the uncertainty of the depth map. The parameter ζ is intended to increase the weight of the inside region to counter the minimal surface bias introduced by unreliable depth maps.

Given those regional terms, I compute the optimal f using the Jacobi method. This results in an evolution of f analogous to the depth map interpolation in Eq. 5.3:

$$f_X[t+1] := \frac{\mu_q c_q(X) q(X) + \mu_\phi c_\phi(X) \phi(X) + \mu_r \frac{1}{6} \sum_{Y \in N(X)} f_Y[t]}{\mu_q c_q(X) + \mu_\phi c_\phi(X) + \mu_r}. \tag{5.20}$$

Here, $N(X)$ is a von Neumann neighborhood around voxel X, i.e. its six closest neighbors at most (fewer on the edges of the voxel grid where some are missing).

Finally, I extract a watertight mesh from f by applying the marching cubes algorithm [22].

5.4 Experiments

I have performed the proposed reconstruction procedure on the sparse depth maps obtained through the method presented in the previous chapter. A number of dense depth maps are shown in Fig. 5.1 and the reconstructed surface is shown in Figs. 5.2 and 5.3.

Figure 5.1: Depth map interpolation. **Top row:** the reference views; **center:** sparse depth maps; **bottom:** interpolated depth maps. Note that the noise in the uncertain areas disappears while the depth values near the edges remain constant.

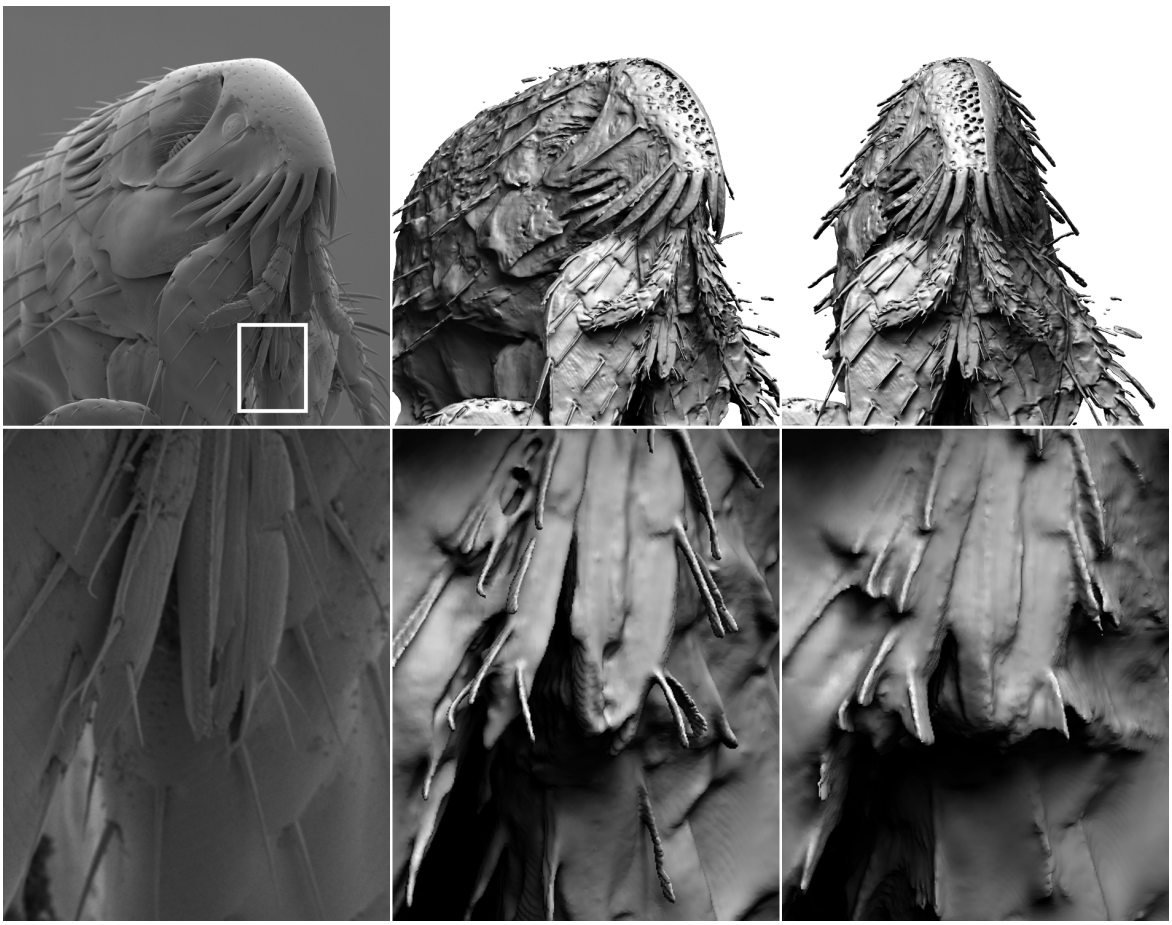


Figure 5.2: Reconstruction of a cat flea. Top, from left to right: one of the input images; two views of the full reconstruction. Bottom: enlarged area of the input image; a reconstruction obtained through the full quadric based method; a reconstruction based only on the depth maps (i.e. $\mu_q = 0$ and $c_\phi(X) = 1, \forall X$ in Eq. 5.12).

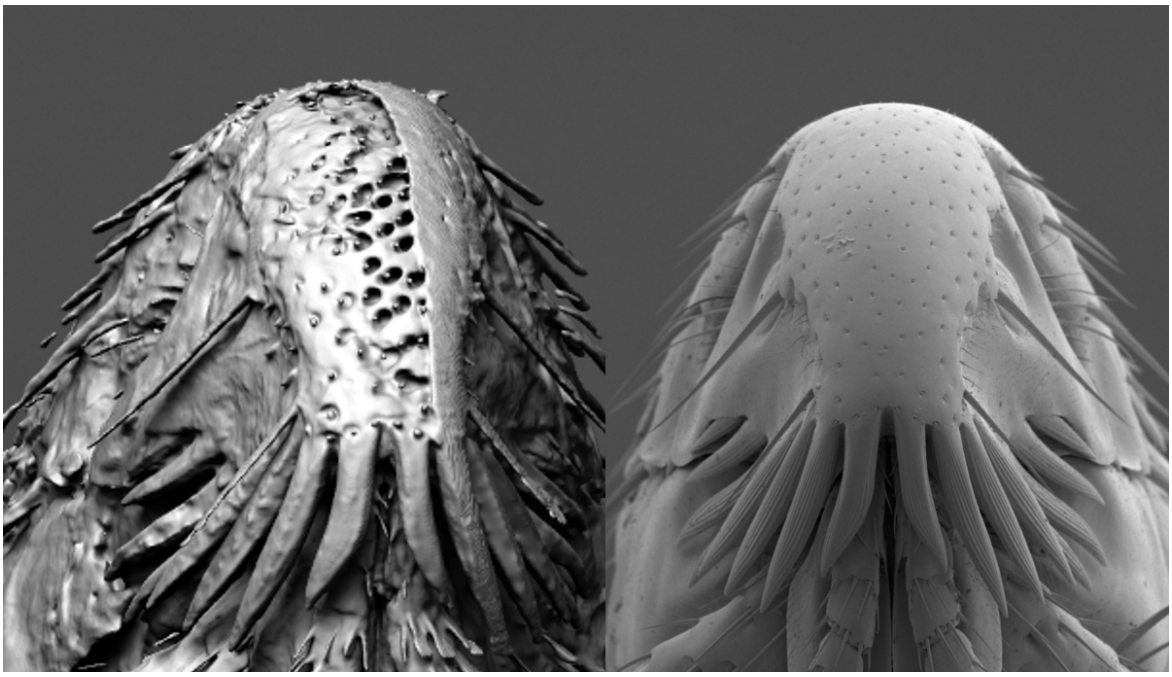


Figure 5.3: Reconstruction of a smoothly curved area. The surface is only reconstructed correctly on the right hand side, where it is seen as a contour in part of the sequence. Smooth areas that are only seen frontally do not allow for a reliable reconsturction using the presented method.

5.5 Conclusions

As can be seen in the figures, the method allows for the reconstruction of very fine untextured features. At the same time, the quadric-based reconstruction fails in smooth untextured regions unless they are seen as contours in some of the images. One reason for this is that spurious edges observed in those areas lead to arbitrary quadrics. In the next chaper, I will describe a reconstruction method that works on a wider range of viewing angles and that no longer relies on a fragile quadric model to extrapolate the curvature of curved contours.

Chapter 6

Photoconsistency-Based Reconstruction from Image Grids

In the previous two chapters, I have shown that dense sequences of secondary-electron images allow for the reconstruction of very fine surface features. This was possible because under sufficient angular resolution, the gray value of a surface point changes only marginally from frame to frame. That angular resolution can, however, only be obtained at the cost of angular coverage.

In the following chapter, I will describe a method to reconstruct such shapes from images that have been captured under a wider range of viewing angles. In this new scenario, the images no longer take the form of a sequence, but instead that of a grid. A scanning electron microscope allows us to **rotate** the probe around the frontal axis and to **tilt** it around the horizontal axis. Since the horizontal axis is itself a function of the current rotation angle, this allows us to view the probe from different sides at different tilts.

By recording a tilt sequence at different rotation angles, it is possible to obtain a grid of different rotation and tilt combinations. In my setup, this grid corresponds topologically to a cylinder. While it is cyclical around the rotation direction, the tilt angles only reach down to a maximum angle of 60°. Although this is a specific limitation of the equipment that was used, as long as in-plane rotations are neglected, such a grid will always be two-dimensional. An intuitive way to imagine this is to map the viewing directions onto points on a sphere. In my specific case, those directions are all contained within a $\pm 60°$-cone around the north pole.

As was discussed at the beginning of chapter 5, such an angular coverage can only be reasonably obtained at the cost of angular resolution. For that reason, a denoising depth estimation in the manner presented in chapter 4 is no longer possible. Instead, the depths have to be estimated in a less reliable way, and a surface reconstruction technique has to be applied that is robust to those less reliable depth maps.

While in the case of image sequences, I used secondary electron (SE) images exclusively, in this case, I recorded both secondary and backscattered electron (BSE) images. In the following chapter, the photoconsistency is still only measured from SE images, since their gray values change far less under a changing viewing direction. The BSE images are only considered during the depth map interpolation process, because they indicate edges in places where the corresponding edges in the SE images are very faint.

The method that I am proposing exploits the anisotropic total-variation (TV) surface re-

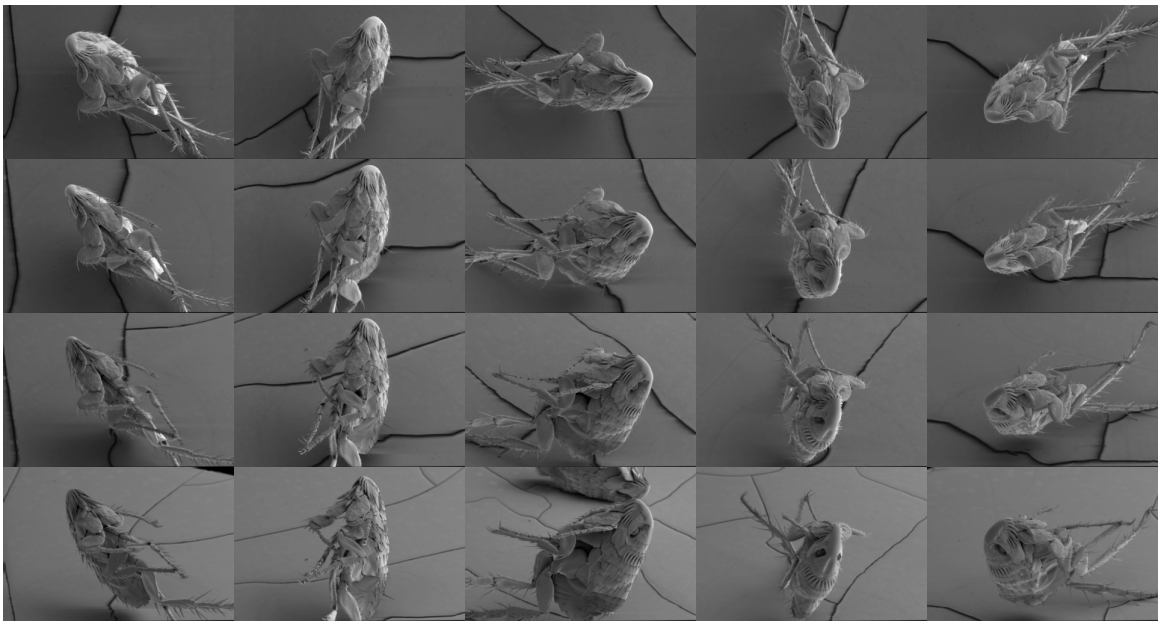


Figure 6.1: Example of an image grid showing five rotation angles (columns) and four tilt angles (rows).

construction framework presented by Kolev et al. [41], and it uses the efficient primal-dual saddle-point algorithm that has been described in that paper.

Formally, the surface reconstruction is accomplished by minimizing the following energy functional for the indicator function $u : \mathbb{R}^3 \mapsto [0, 1]$:

$$E_{\mathrm{TV}}(u) \coloneqq \int_{\Omega} -f(X)u(X) + |D(X)\nabla u(X)|dX, \tag{6.1}$$

where $f : \mathbb{R}^3 \mapsto \mathbb{R}$ is the regional term while the anisotropic regularizer D takes the form of a symmetric, positive definite 3×3-matrix. The scalar μ provides a weighting of the regional term vis-à-vis the regularizer. This formulation is equivalent to the one used in previous work [41, 80].

The specific regularization tensor, as well as the regional terms that I use, are novel and pursue different aims from the ones proposed in the original formulation. My regional term has to be computed in a way that makes it robust to the specific problems that arise in conjunction with sparsely textured curved surfaces and thin protruding features, while the regularizer pursues an altogether novel approach.

Unlike the original formulation, my regularizer does not assume that the local surface normal is known a priori. The estimation of such a normal is only possible if surface patches of a certain size are considered. Similarly to the use of image windows to estimate image depths that has been discussed at the beginning of chapter 4, such a surface patch can be contaminated by points from different surfaces.

For that reason, my regularizer does not assume that the surface normal can even be known from the local area of a point alone. Specifically, when looking at an edge on the surface from multiple angles, that edge only tells us one of the two dimensions that make up the surface normal. It tells us that the normal is certainly perpendicular to the edge, but this only limits the space of possible normals to a circle. An additional nearby edge is necessary to estimate the precise normal.

It is also interesting to note that the edges that we see on surfaces are often hinges, i.e. they are formed by a local change in surface orientation. Although the surface exhibits two different normals on the two sides of such a hinge, both of the normals have to lie on the same circle perpendicular to it.

My proposed regularizer accounts for these facts, and it only assumes that minimal amount of information that can be extracted from the images locally. The actual surface normal then emerges as part of the optimization process. This will be explained in more detail later in this chapter.

My algorithm begins with the estimation of image depths in way that is robust to changes in surface radiance. These sparse depths are then interpolated using the method presented in chapter 5, which are used in conjunction with the sparse depths to construct the regional term f. The regularizer D is computed from the sparse depths alone.

6.1 Depth Estimation Under Changing Radiance

Depth estimation under a wider baseline angle confronts us with two challenges. First, the number of images is insufficient for the occlusion-robust denoising approach that I have presented for narrow-baseline sequences. Instead, in order to remain robust to occlusion, the estimation has to be performed on the bare minimum of neighboring views, i.e. the reference view and its two closest neighbors.

Second, in the absence of significant denoising, a single pixel no longer carries sufficient information for a reliable estimate of depth. In order to still avoid the use of explicit patches, I apply a minimalistic form of local cost volume filtering [81] instead.

As I have mentioned in the introduction to this chapter, only SE images are used for depth estimation. The lightness changes seen in the BSE images are far more pronounced, so they are inadequate for a photoconsistency-based depth estimation approach. In order to deal with the lightness changes that are nevertheless present in the SE images, I first compute high-pass filtered images $u_i = v_i - G_\nu * v_i$, where G_ν is a convolution with a Gaussian with a standard deviation of ν. The depth estimation is then performed through cost volume filtering with a minimalistic 5×5-pixel Gaussian kernel and the following pixelwise matching cost for a point in space X:

$$C(X) \coloneqq \min_{a,b} \sum_i (u_i(X) - a)^2 + |v_i(X) - b|, \tag{6.2}$$

$$= \sum_i (u_i(X) - \mu(X))^2 + |v_i(X) - m(X)|, \tag{6.3}$$

where $u_i(X)$ is a shorthand for the value of image u_i at the point to which X projects, $\mu(X)$ is the mean of all $u_i(X)$ and $m(X)$ is the median of all $v_i(X)$. This cost can be understood as follows. It is an L^2 cost over the high-pass filtered images u_i, but, since the u_i do not contain all information from the v_i, using u_i alone can hallucinate false matches that can outweigh the true matches. The second summand is therefore intended as a tie-breaker that will select the depth at which there is more agreement between the v_i, in cases where the u_i happen to agree at multiple depths.

This cost $C(X)$ is evaluated at a set of discrete points X_z arranged along the viewing ray for every pixel, yielding a set of costs $C_z(x, y)$. Next, cost volume filtering is perfomed by Gauss-filtering the images $C_z(x, y)$ corresponding to each depth z, resulting in the filtered

costs $\bar{C}_z(x, y)$. These are then mapped onto probabilites $p_z(x, y) = \exp(-\bar{C}_z(x, y)/\beta)$, where β is a sharpening parameter. The probabilites are then normalized along each ray. The depth z^* of maximal p_z is then computed by quadratic interpolation around the discrete maximum, analogous to Eq. 4.13. In spite of using cost volume filtering, those depths are still sparse because the filter kernel is very narrow. The purpose of the filtering is merely to make the depth estimates more reliable, not to fill them. The probability $p_{z^*}(x, y)$ itself is used as the corresponding confidence, $c_i(x, y)$. In contrast to the narrow-baseline scenario, the variance is no longer a reliable measure of confidence.

Analogously to the method by Yücer et al. [21], the depths are then filtered as follows. The 3D points corresponding to the estimated depths are forward-projected into the voxels of a volume, where the confidences of all the depth maps are summed up. Only points that project into voxels with a sufficient confidence sum are kept, while the confidences of the others are set to zero. This eliminates most of the spurious matches.

Finally, interpolated versions $z^{(D)}$ of the sparse depth maps z^* are computed as described in section 5.1. Since no denoised images are available to tell us the locations of the contours, the raw input images v have to be used in their place. The increased noise level in these images demands a larger value for the contrast parameter λ in Eq. 6.4, which in turn further aggravates the problem of dislocated thin features.

To counteract this, both the SE and the BSE images are used to compute the pairwise pixel affinities a_{pq} in eq. 5.2,

$$a_{pq} \coloneqq \exp\Big(-\frac{1}{\lambda_S^2}(v_{S,p} - v_{S,q})^2 - \frac{1}{\lambda_B^2}(v_{B,p} - v_{B,q})^2 \Big), \tag{6.4}$$

where v_S and v_B are the pixel values of the SE and BSE images respectively. Using both images increases the chance of observing an image edge at the location of a contour. Failure to find such an image edge leads to effects of depth leakage — smooth protruding features, both thin ones and large, smoothly curved ones, are dislocated towards the background.

6.2 Anisotropic Regularizer

The sparse depth maps z^* obtained in the previous section already contain the depths of most of the image edges of significant contrast. Since my anisotropic regularizer is based on the observed image structure, these sparse depths and the original input images are all that is necessary to compute the regularizer matrix $D(X)$.

In the following, I will first explain the reasoning behind the regularizer in more detail, and then describe how it is computed.

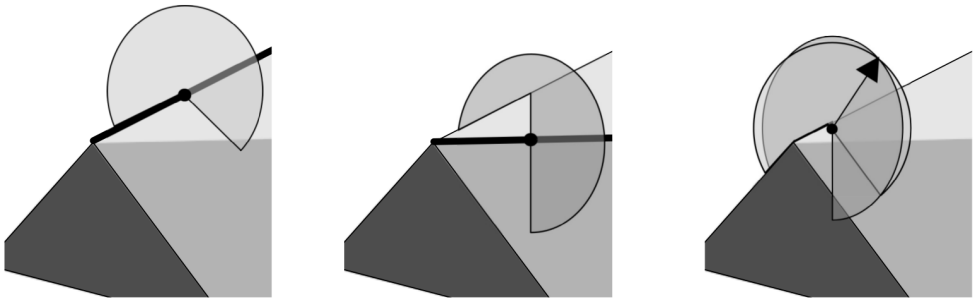
6.2.1 Reasoning

My goal is to reconstruct a surface from image edges as the largest feature. As mentioned previously, considering image areas larger than a single pixel exposes us to the risk of missing small shape features. Since the wide-baseline scenario no longer allows us to use single pixels as our largest features, single local edges are the next larger choice.

Each time we observe an edge in an image, we can ask ourselves to which curve in 3D space that edge corresponds. Since we are working with images of finite resolution, we can reduce

every observed edge locally to a straight 2D line by looking at the image gradient. The curve we seek must thus be the 3D line in which all the planes intersect that are formed by back-projecting corresponding 2D lines into 3D space.

Next, we ask what that 3D line tells us about the surface around it. Since the edge has to be contained in the surface, we know that the direction of the line can have no component in the direction of the surface normal — otherwise, the line would leave the surface. The surface normal thus has to lie within the plane perpendicular to the line direction. If we also assume that the normal is of unit length, then this means that the normal lies on a circle around the line. This is illustrated in fig. 6.2.



(a) hinge (b) surface edge (c) between edges

Figure 6.2: Illustration of the above argument. Each edge limits the space of possible normals to a circle (a, b). A single normal can only be estimated if multiple nearby edges are considered (c).

Such a surface line can be formed either by a change in surface color or by a change in surface orientation, i.e. a hinge edge. While in the former case, the line merely provides insufficient information to reconstruct the surface normal, in the latter case, that information does not even exist. Around a hinge edge, the surface has two different normals on either side of the line, though both of them have to be perpendicular to it.

From this we conclude that in the context of reconstruction from edges, the surface normal can only be determined from multiple surrounding edge observations. From the edge, the minimal feature itself, we can only determine a circle on which the normal has to lie. This is the essential assumption behind my regularizer. I do not presume to know the normal beforehand, but I only enforce consistency with the line observations. This allows the unknown normal to emerge from the bottom up in the smooth areas between the edges. Although this observation was already made by Ikeuchi [56], it does not form the basis for current anisotropic reconstruction methods [41, 80]. Both of these methods make explicit normal assumptions.

It should be noted at this point that our concept of a discrete edge is an abstraction. In reality, image edges take up multiple pixels across their width, and the gradient is usually stronger in the middle than at the ends. Also, hinge edges are in reality only areas where the surface exhibits strong curvature in one direction — an infinitely sharp hinge cannot exist in nature. Still, in both cases it remains true that the surface normal has to be perpendicular to the line implied by the image gradients. Thus, in order to deal with curved geometry, I do not perform any edge-thinning operations, but instead use all the gradients.

6.2.2 Structure Tensor Computation

My anisotropic regularization tensor is computed from the equivalent of a 3D structure tensor, in a similar manner as one would compute an anisotropic diffusion tensor from the structure tensor [43, 44], e.g. for the purposes of denoising.

Traditionally, a 2D structure tensor $J(x, y)$ of an image $u(x, y)$ is computed by first evaluating what I will refer to as the edge tensors $E(x, y) = \nabla u \nabla u^T$ at every pixel (x, y), where ∇u is the image gradient [82]. The 2×2 matrices E are symmetric and they have one positive eigenvalue corresponding to ∇u as eigenvector, while the other eigenvalue is zero.

Then, for a traditional structure tensor, $E(x, y)$ is filtered, resulting in the structure tensor $J(x, y) = G_\rho * E(x, y)$, where G_ρ denotes a convolution with a Gaussian kernel with a standard deviation of ρ.

The new matrices $J(x, y)$ are now positive semidefinite. In areas where edges of different orientations meet, J has two large eigenvalues. In areas showing only one dominant orientation, it has one large eigenvalue and its eigenvector points in the dominant gradient direction. Finally, in smooth areas, both eigenvalues of J are small or zero.

In the case of a 3D structure tensor, such as those that can be obtained from 3D CT images, the edge tensors $E(x, y, z)$ still only have one non-zero eigenvalue. This is because they are still outer products of a vector with itself, albeit a 3D vector. The structure tensor $J(x, y, z)$, however, can now have up to three large eigenvalues, allowing us to distinguish between smooth, planar, line-like and corner-like areas.

Assuming we know the depth of a large number of image edges, my 3D structure tensor is computed as follows. Let $\nabla u_i(x, y) = (u_x, u_y)^t$ be the gradient of image i at (x, y). As mentioned above, any gradient at any given point (x, y) implies a 2D line $l(x, y, 1)^t = 0$. The row vector $l = (u_x, u_y, d)$ is composed of the image gradient and an offset d that describes the position of the line. It is given by $d = -(xu_x + yu_y)$. We lift the line into 3D space by finding the plane $\pi X = 0$ given by the row vector π that corresponds to line l. From $(x, y, 1)^t = T_i X$ and $l(x, y, 1)^t = 0$, it follows that $\pi = lT_i$.

Next, we define a partial 3D gradient vector

$$g = |\nabla u| \frac{\pi_{xyz}}{|\pi_{xyz}|}. \tag{6.5}$$

This vector points in the direction of the plane normal, and its length is equal to that of the gradient. We treat g as if it were the gradient of a 3D image, and we use it to compute the partial edge tensor $F_i(x, y) = gg^T$.

Since we assumed the depths of edge pixels to be known, we now use that depth to project this symmetric 3×3 matrix F into 3D space. At every voxel (x, y, z), we compute the 3D edge tensor,

$$E(x, y, z) = \frac{\sum_{i \in \mathcal{S}} c_i F_i}{\sum_{i \in \mathcal{S}} c_i}, \tag{6.6}$$

where $\mathcal{S}$ is the set of all pixels in all images that project onto voxel (x, y, z), and c_i are the depth confidences of those pixels.

Because we have added F_i from multiple pixels and images, the $E(x, y, z)$ are not all rank 1 matrices. On the contrary, if a voxel contains a surface line that is seen in more than one image, then its E will indeed show two larger eigenvalues and a small or zero one, with the latter corresponding to the line direction as eigenvector. This follows from the fact that all

partial gradients g, i.e. the normals of all planes containing the line, have to be perpendicular to the line. Areas with only one dominant eigenvalue do occur within smooth areas, though, if those areas are seen near the contour in any of the images.

Next, the matrix field E is filtered with a Gaussian of variance σ_E, and the resulting pseudo structure-tensor J is used to classify (in a continuous sense) the underlying surface structure.

6.2.3 Regularization Tensor Construction

My regularization tensor is inspired by the work by Mendrik et al. [44], who proposed an anisotropic diffusion process as a means of denoising 3D computer tomography (CT) data. They developed a diffusion tensor that selectively performs different types of diffusion, depending on the local environment of each voxel in a CT image. If the local image information suggests a thin, line-like feature, then the diffusion only takes place along that line, while in the case of a planar environment, the image is diffused in both directions parallel to that plane. In the extreme case of a uniform environment, the diffusion is isotropic, and in the case of point-like features, there is no diffusion.

Their approach has been termed *hybrid diffusion with a continuous switch* (HDCS), because the distinction between those four environments is not a discrete classification, but instead, the diffusion process is able to interpolate smoothly between the four behaviors. I chose not to adopt their HDCS formulation directly, since it is aimed at CT image denoising and its four different parameters do not map well to the scenario of incomplete edge information. Also, their formulation is built on top of the (originally 2D) edge-enhancing diffusion (EED) and coherence-enhancing diffusion (CED) filters proposed by Weickert [42, 43], and it carries those parameters along with it.

Instead, my regularization tensor definition aims to be optimally comprehensible, and is given as follows.

The pseudo structure-tensor J is subjected to an eigenvalue decomposition, yielding three positive eigenvalues λ_i and eigenvectors v_i.

$$J = (v_1, v_2, v_3)\Lambda(v_1, v_2, v_3)^t. \tag{6.7}$$

where Λ is a diagonal matrix with λ_i as its diagonal entries.

The type of environment is then inferred from the eigenvalues alone. The environment type is expressed by four scalar values $\in [0, 1]$ to which I will refer as the four *qualities*: the void quality q_V, the line quality q_L, the plane quality q_P and the corner quality q_C. The four values are required to sum to one.

Since lines form the central concept in the derivation of my regularizer, line-like features are considered the default environment. For that reason, the other three qualities are subtracted in a given order, so that the remaining number is the line quality q_L. The reader may find it helpful to think of this classification as a sequence of areas that are chipped away from the 3D volume which is spanned by the three eigenvalues λ_i.

In the following, we can assume without loss of generality that the λ_i are given in descending order, with λ_1 being the largest eigenvalue.

First, we define the void quality:

$$q_V := \exp(-\frac{\lambda_1^2}{\tau_V^2}). \tag{6.8}$$

This ensures that q_V will generally be zero, unless even the largest eigenvalue is sufficiently small compared to the parameter τ_V: this means that no edges are seen in that region. This is the only definition that considers the absolute values of the λ_i. All following definitions are invariant under uniform scaling of the eigenvalues, which makes them independent of the number of observations of a given surface area.

Next, the corner quality is defined as

$$q_C \coloneqq (1 - q_V)(\frac{\lambda_3}{\lambda_1})^{k_C}, \tag{6.9}$$

where k_C is another parameter describing the strictness of corner classification. The corner quality can only approach 1 if q_V is close to zero and all three eigenvalues are of similar magnitude.

Next, the plane quality is given by

$$q_P \coloneqq (1 - q_V - q_C)(\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3})^{k_P}, \tag{6.10}$$

where k_P is the corresponding strictness parameter for plane classification. The plane quality is only close to 1 if the two smaller eigenvalues are of similar size, i.e. significantly smaller than the largest one.

Finally, the line quality is given by

$$q_L \coloneqq 1 - q_V - q_C - q_P. \tag{6.11}$$

With the four qualities defined, we can now construct the regularization tensor by replacing the eigenvalues λ_i by new ones, μ_i,

$$\mu_1 \coloneqq \epsilon + (1 - \epsilon)\ q_V \tag{6.12}$$
$$\mu_2 \coloneqq \epsilon + (1 - \epsilon)(q_V + q_P) \tag{6.13}$$
$$\mu_3 \coloneqq \epsilon + (1 - \epsilon)(q_V + q_P + q_L) \tag{6.14}$$

and then recomposing the matrix,

$$D \coloneqq (v_1, v_2, v_3)\Lambda_\mu(v_1, v_2, v_3)^T, \tag{6.15}$$

where Λ_μ is a diagonal matrix with μ_i as its diagonal entries, and $\epsilon > 0$ is a parameter of small value that ensures that D is strictly positive definite.

6.3 Regional Term

The purpose of the regional term $f(X)$ is to bias certain voxels to be classified as either inside or outside, depending on the sign of f. In the following, I will assume that a positive sign means inside. The regional term is the only active component of this surface reconstruction framework. This means that without it, an empty space would always form the optimal surface, because the regularizer favors a smaller surface area and the area of an inexistent surface is effectively zero.

The main problem in the construction of such regional terms is the fundamental asymmetry of vision: we can only ever observe the space up to the surface of an object, but we never

have any observations of the volume within it. Thus, while seeing a point in space in front of a depth map is evidence of that point being outside the object, there is never any evidence for a point being inside.

Traditionally, surface reconstruction methods would construct regional terms under the assumption of a locally planar surface. Then, if a point in space is located behind a given depth map, that observation can be taken as evidence of the point being inside the object. In order to prevent observations from opposite sides of the object from interfering with each other, this assumption is usually only assumed valid if the distance between the depth sample and the point is small enough [83].

Since we are dealing with very thin features, that critical distance would have to be equally small. In that case, the bulk of the inside volume could not be classified at all, i.e. f would be zero there. Only a small number of erroneous observations would then suffice to collapse large parts of the inside volume to an erroneous outside classification. If such an unstable regional term f were to be constructed from the sparse depth maps and their confidences, then the collapsed shape would assume the skeletonized form that has been mentioned briefly in 5.1.

Although this could be counteracted through the use of an inflationary (or, ballooning) term [24] that blindly biases each point towards the inside (i.e. a constant positive offset on f), such a term would also inflate all smooth surface areas, since they do not offer any evidence of being inside or outside either.

For these reasons, I conclude that interpolated, dense depth maps without any notion of confidence are necessary to compute a regional term that exhibits stable behavior. If ideal dense depth maps, free of errors and noise, were available, then computing f from them would be trivial. Every point in space that is visible in front of even a single depth map could be immediately classified as outside, while only points that are occluded by all depth maps could potentially be inside the object.

In the real scenario, however, the estimated depth maps $z^{(D)}$ suffer from three specific degradations:

1. Faint thin features are dislocated towards the background.
2. Curved protruding areas are flattened.
3. The interpolated depth maps form arbitrary sheets across smooth image regions around contours.

The first two degradations have been discussed at the end of section 5.1. They are generally even worse in the wide-baseline case, because the depth maps are of lower quality. They are destructive, i.e. they remove parts of the object, resulting in points within the object that are nevertheless seen in front of a number of depth maps. At the same time, surface regions located within cavities are really only visible in a small number of images. Allowing points that are seen in front of a certain number of depth maps to still be classified as inside would also start filling up those cavities. The third type of degradation aggravates this problem further, because it tends to fill in cavities as well.

In order to deal with these problems, I have chosen the following approach. Only points that are occluded by almost all of the depth maps are assumed to be certainly inside, and only points that are visible in front of a very large number of depth maps are assumed to be certainly outside. Points that are seen by an intermediate number of views are handled by three specific terms that correspond to the three types of degradations.

Formally, let

$$f \coloneqq \alpha_c(f_{\text{in}} - f_{\text{out}}) + \alpha_1 f_1 + \alpha_2 f_2 - \alpha_3 f_3, \tag{6.16}$$

where f_{in} and f_{out} are the aforementioned certain terms, while $f_{1,2,3}$ are the terms designed to deal specifically with those degradations. The positive scalars $\alpha_{1,2,3}$ control the relative weighting of the terms. The terms $f_{1,2,3}$ will be defined and explained in the following.

6.3.1 Certain Terms

Let $w_D(X)$ be the dense visibility function that counts the number of views that see point X in front of their respective dense depth map $z^{(D)}$. This function is generally small inside the object, though degradations of type 1 and 2 tend to increase its value within thin features and underneath smoothly curved convex surface areas. Outside of the object, w_D can also be very small within cavities, and it is maximal in areas that are seen by the most views. Slices through such a dense visibility function w_D can be seen in Figs. 6.6, 6.7 and 6.8 on pages 74, 75 and 76 respectively.

Practically, w_D is computed by projecting the position of each voxel into every depth map and then comparing the depth of the voxel to that of the corresponding depth map pixel. If the depth of the voxel is smaller than the value in the dense depth map, then the value of the voxel is increased by one.

The certain terms are then defined as follows:

$$f_{\text{in}} \coloneqq 1 - \exp\Big(- \frac{1}{\sigma_{\text{in}}^2} \max(0, \tau_{\text{in}} - w_D)^2 \Big), \tag{6.17}$$

$$f_{\text{out}} \coloneqq 1 - \exp\Big(- \frac{1}{\sigma_{\text{out}}^2} \max(0, w_D - \tau_{\text{out}})^2 \Big), \tag{6.18}$$

where τ and σ are the respective inside and outside thresholds and tolerances. Their values can be determined by looking at cross-sections of w_D and finding the smallest value that does not occur outside the object (τ_{in}) and the greatest value that does not occur inside of it (τ_{out}).

6.3.2 Thin Term

The thin term f_1 represents the thin features that are missing in many of the depth maps. Because many depth maps see through such features, the value of w_D inside of them is significantly greater than zero. Because of the depth maps that do contain such features, that value of w_D is still smaller than in their immediate neighborhood. At the same time, these features generally form edges in the images, so they also often exhibit a large sparse confidence $c_i(x, y)$.

The idea behind the thin term f_1 is to find a local threshold value θ_1 that indicates the value of w_D at the surface of the features. Then, the difference $\theta_1 - w_D$ is positive inside and negative outside of them.

To compute f_1, we first need to construct an indicator of the position of the true surface. This is done by computing a non-maximum-suppressed edge tensor E_s from the sparse depths maps, analogous to the edge tensor E in Eq. 6.6. For E_s, we only consider gray value edges that exhibit maximal intensity along their gradient direction. This is the same non-maximum suppression operation that has been applied to the depth values for the contour candidates

in 5.2.1. The edge tensors are not subject to the degradations introduced through depth interpolation, because they are computed from the sparse depth maps.

We then compute the largest and second-largest eigenvalues of E_s, e_1 and e_2. While e_1 can be large even on smoothly curved surfaces, i.e. the locally plane-like environments described in 6.2.2, e_2 is only large in a line-like environment. Such an environment is given either by a surface line or hinge, or by the contours of a strongly curved feature. Both indicators are undisturbed by the degradations that are introduced through depth interpolation. Since we are only interested in thin features here, only e_2 is used for f_1. The smoothly curved surfaces that are contained in e_1 will be dealt with by f_2.

The thin term f_1' is now computed as the signed difference $\theta_1 - w_D$, scaled by the local average of e_2, while θ_1 is itself defined as the e_2-weighted local average of w_D:

$$f_1' \coloneqq (G_\sigma * e_2)(\theta_1 - w_D) \tag{6.19}$$

$$= (G_\sigma * e_2)(\frac{G_\sigma * (e_2 w_D)}{G_\sigma * e_2} - w_D) \tag{6.20}$$

$$= (G_\sigma * (e_2 w_D)) - (G_\sigma * e_2) w_D. \tag{6.21}$$

The effect of type 1 degradations is purely destructive, so we are only interested in points where f_1' is positive. We thus subject it to a smooth thresholding operation,

$$f_1 \coloneqq 1 - \exp\Big(- \frac{1}{\sigma_{f1}^2} \max(0, f_1')^2 \Big). \tag{6.22}$$

The final f_1 is now ≈ 1 where f_1' is significantly larger than the parameter σ_{f1}, which is set to a very low value (in my experiments, σ_{f1} was set to 10^{-3}).

6.3.3 Curved Term

The purpose of the curved term f_2 is to reconstruct smoothly curved areas that are missing in many depth maps. Because of their smoothness, these areas generally do not allow for a reliable estimation of depth, and they are thus often flattened to the depth of their contours in the interpolation process. As a result, the value of w_D is larger than zero underneath the surface of such areas. Just as in the case of the missing thin features, some depth maps still see these features correctly, so there is generally a visible strong gradient in w_D along the true surface.

Unlike the thin features, the effects of degradations of type 2 are not strictly local. For a sufficiently large smoothly curved feature, the area where the value of w_D is excessive can reach deep inside the object. Without f_2, our regional term would be at risk of subsurface collapse, i.e. a gap could form between the deep inside region indicated by f_{in} and the strictly locally determined inside region indicated by f_1.

To prevent this, the term f_2 serves to indicate whether the local value of w_D is greater or smaller than θ_2, the value of w_D at the nearest known surface point. If the value is smaller, then that means that the point lies beneath the surface.

For smoothly curved areas, the indicator of surface points is defined as follows. We define another scalar field $c_s(x, y, z)$ as the local density of points from the sparse depth maps, scaled by their respective confidence. It is computed as the sum of the sparse confidences $c_i(x, y)$ of all sparse depths from all the images that back-project into each voxel. This

forms a diffuse cloud around the true surface, and, like e_1 and e_2, it is also not influenced by the degradations of the depth interpolation process. That diffuse point field is further sharpened by multiplying it with the gradient length of w_D, resulting in the sharp point field $c_p = c_s|\nabla w_D|/N_i$, where N_i is the number of images. Finally, we compute the sum $m = e_1 + c_p$, yielding the surface indicator m.

In order to compute f_2, we first need to determine the gradient vector flow (GVF) [84] of w_D. This is a vector field that corresponds to the gradient ∇w_D where that gradient is large, and that is smooth otherwise. It has been developed in the context of local segmentation methods, where it has been shown to prevent the locally evolving solution from getting stuck in smooth regions.

Formally, the gradient vector flow is defined as the vector field $h_0 : \mathbb{R}^3 \mapsto \mathbb{R}^3$ that minimizes

$$E_{\mathrm{GVF}}(h) \coloneqq \int_\Omega |\nabla w_D(X)|^2 |h(X) - \nabla w_D(X)|^2 + \mu_{\mathrm{GVF}} \sum_{i \in \{x,y,z\}} |\nabla h_i(X)|^2 dX. \tag{6.23}$$

I have performed this optimization using another Jacobi method, analogously to the surface optimization in Eq. 5.20 on page 53:

$$h_{i,X}[t+1] \coloneqq \frac{|\nabla w_D(X)|^2 (\nabla w_D(X))_i + \mu_{\mathrm{GVF}} \frac{1}{6} \sum_{Y \in N(X)} h_{i,Y}[t]}{|\nabla w_D(X)|^2 + \mu_{\mathrm{GVF}}}. \tag{6.24}$$

We are only interested in the direction of h_0, so we normalize it safely,

$$h(X) \coloneqq \frac{h_0(X)}{|h_0(X)| + \epsilon}. \tag{6.25}$$

Using this field h, we can now propagate any value v from the surface, where $m > 0$, along the flow by finding a scalar field w that minimizes

$$E_{\mathrm{P}}(w, v) \coloneqq \int_\Omega m\Big(w(X) - v(X)\Big)^2 + \Big(h(X) \cdot \nabla w(X)\Big)^2 dX. \tag{6.26}$$

Here, the last summand minimizes the slope of w along h. This serves to propagate the value of v from the surface along the flow field.

In this anisotropic case, the Jacobi method has proven too unstable, so the optimal w is evolved using a gradient descent method instead:

$$w(X)[t+1] \coloneqq w(X)[t] + \delta_t \Big(2m(v(X) - w(X)[t]) + \mathrm{div}(D_h(X) \nabla w(X)[t])\Big), \tag{6.27}$$

$$D_h(X) \coloneqq h(X) h(X)^t. \tag{6.28}$$

At points where the data weight m is zero, this process is equivalent to anisotropic diffusion along the rank 1 diffusion tensor D_h.

We use this propagation energy to compute the local surface averages of $G_{\sigma_2} * w_D$ and m itself:

$$w_D^{(2)} \coloneqq \operatorname*{argmin}_w \Big(E_{\mathrm{P}}(w, G_{\sigma_2} * w_D)\Big) \tag{6.29}$$

$$m^{(2)} \coloneqq \operatorname*{argmin}_w \Big(E_{\mathrm{P}}(w, m)\Big). \tag{6.30}$$

The field $m^{(2)}$ indicates whether there is a surface point nearby along h, and if there is, then $w_D^{(2)}$ indicates the value of $G_{\sigma_2} * w_D$ at that point.

Analogously to the definition of f_1', we compute f_2 as

$$f_2 \coloneqq m^{(2)}(\theta_2 - w_D) \tag{6.31}$$

$$= m^{(2)}(\frac{w_D^{(2)}}{m^{(2)}} - w_D) \tag{6.32}$$

$$= w_D^{(2)} - m^{(2)} w_D. \tag{6.33}$$

Just like f_1', f_2 is positive if the closest surface point exhibits a greater value of w_D, and negative otherwise, while its magnitude indicates its confidence. Unlike degradations of type 1, those of type 2 can be both constructive and destructive. For that reason, both the positive and the negative range of f_2 are required.

6.3.4 Erosion Term

Unlike degradations of types 1 and 2, those of type 3 are always constructive, meaning that they add matter where there should be empty space. They occur around contours of features that are seen in front of an untextured background and can be understood as foreground depth leaking into the background. If the background is untextured, inferring its depth from photoconsistency is not possible. As a consequence, if the contours seen in the images are not strong enough, the depth of background pixels is erroneously inferred from the nearest foreground features, which leads to the formation of sheets around those features.

The corresponding regional term f_3 is thus destructive. Its value is non-negative, and it appears with a negative sign in eq. 6.16. It is computed from observations of surface points seen behind a given point in space in the sparse depth maps.

In order to define f_3, we first define a scalar field $w_S(X)$ similarly to $w_D(X)$ as the sum of the confidences of all points in the sparse depth maps that are seen behind X. Practically, $w_S(X)$ is computed by forward-rasterizing the rays corresponding to all points in the sparse depth maps into a volume. The value of all voxels in the volume is initially set to zero. For each voxel that is touched in the rasterization process, the sparse confidence c_i of that depth map pixel is added to the value of the voxel. This rasterization process is necessary for the sparse depth maps, because the sparse points of high confidence might otherwise fall between the sample points of the voxel grid.

From w_S, we can define f_3 as

$$f_3 \coloneqq 1 - \exp(-\frac{1}{\sigma_3^2} \max(0, w_S - \tau_3)^2), \tag{6.34}$$

where σ_3 and τ_3 control the tolerance of f_3 to noise in w_S.

With all the terms computed, the final surface is determined by optimizing eq. 6.1, and extracting the 0.5-isosurface from the resulting scalar field u using the marching cubes algorithm.

6.4 Experiments

I have captured an image grid of the same cat flea specimen that has been used in the narrow-baseline experiments. The specimen was prepared by Ken Goldie, who also helped

me with the use of the FEI Versa 3D Focused Ion Beam (FIB) electron microscope. The ion beam was not used in the experiments so as not to destroy the specimen. The capture itself was done through a script that I wrote for the FEI iFast software which enables the automated capture of large numbers of images.

The image grid consists of 20 evenly spaced rotation angles and 16 tilt angles. The rotation angles cover the full circle, while the tilt angles range from 0° to 60°. An SE and a BSE image were captured from each view simultaneously. Twenty of the 320 SE images are shown in Fig. 6.1 at the beginning of this chapter. The images were captured at a resolution of 6144×2048 pixels.

The grid was calibrated by manually selecting 70 corresponding points in all 320 frames. This extremely time-consuming process was aided by a program that I wrote which tentatively maps points automatically to the nearest frames by considering a small area around the point. Since that automated process is not completely reliable, the user is required to repeatedly flip back and forth between the frames and to correct mismatches. When the user flips between neighboring frames, the program always keeps the on-screen position of the point constant. This allows the user to ascertain that the two points in the two frames really correspond to the same surface point by observing the motion of the surface underneath the selected points. This allows for a calibration at subpixel precision. In spite of my software solution, this calibration process represents the main bottleneck if my method were to be used for any practical applications.

The corresponding points were then used to estimate both the projection matrices and a considerable non-linear lens distortion which was modelled using the very expressive rational function lens distortion model by Claus and Fitzgibbon [85]. The lens distortion was then removed by remapping the images onto images that are consistent with a pinhole camera model.

The calibrated images were used to estimate the radiance-change-robust depth maps described in this chapter, using the two closest frames with the same rotation angle (i.e. $\pm 4°$ tilt). The depth costs and probabilites are visualized in Fig. 6.3 and 6.4. A number of interpolated depth maps are shown in Fig. 6.5. The surface was then reconstructed using the method described in this chapter. The regional terms are shown in Figs. 6.6, 6.7 and 6.8 and the resulting surfaces in Figs. 6.9 and 6.10.

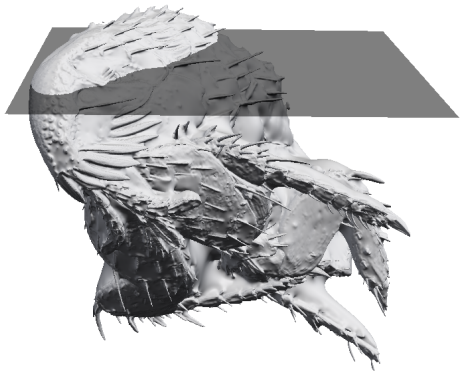
The following parameter values were used:

Depth interpolation:	$\lambda_S = 0.02$	$\lambda_B = 0.02$		
Regularizer:	$\tau_V = 0.01$	$k_C = 2$	$k_P = 10$	
Regional terms:	$\alpha_c = 0.01$	$\alpha_1 = 0.2$	$\alpha_2 = 1$	$\alpha_3 = 0.01$
Certain term:	$\tau_{\text{in}} = 5$	$\sigma_{\text{in}} = 2$	$\tau_{\text{out}} = 230$	$\sigma_{\text{out}} = 100$
Curved term:	$\mu_{\text{GVF}} = 500$			
Erosion term:	$\tau_3 = 0.05N_i$	$\sigma_3 = 0.1N_i$		

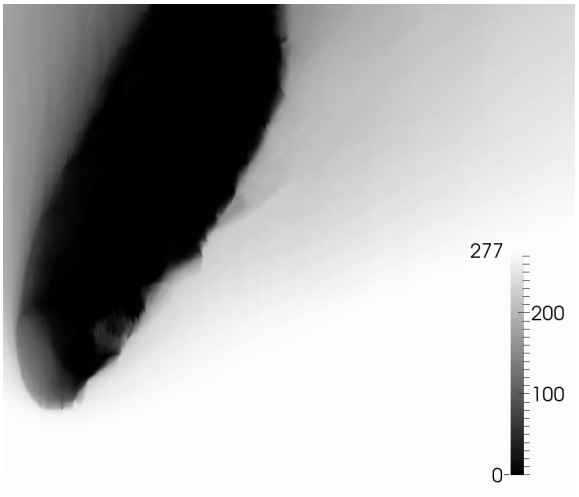
Figure 6.3: Slices through the filtered cost $\bar{C}_z(x, y)$ at the y values indicated by the black lines, corresponding to the energy plots in Fig. 4.3 on page 40. Note the dramatic loss of quality in the wide-baseline scenario. Since only three images are used for the computation, the structure of those images is still visible in the cost slices.

Figure 6.4: Slices through the probabilites $p_z(x, y)$ corresponding to the costs in Fig. 6.3. Note the lower coverage and the increased number of spurious matches compared to the narrow-baseline scenario in Fig. 4.4 on page 41.

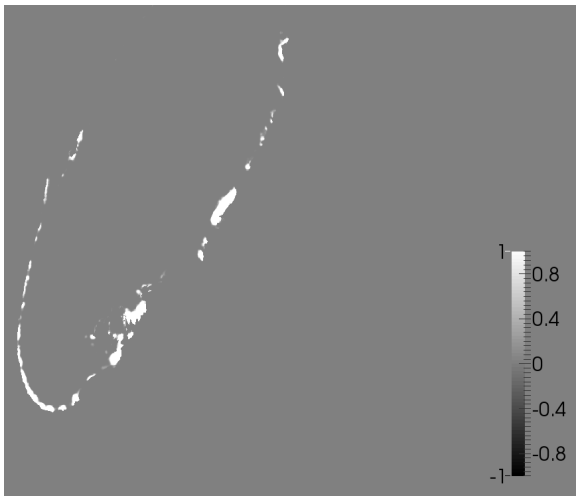
Figure 6.5: Estimated depth maps. **Left:** the input image. **Center:** sparse depth map. **Right:** filled depth map. Note the much noisier appearance of the sparse depth maps compared to those obtained in the narrow-baseline scenario (Figs. 4.5 on page 42 and 5.1 on page 54). The depth values were normalized individually for this visualization, so the gray values cover different ranges in the corresponding depth maps.



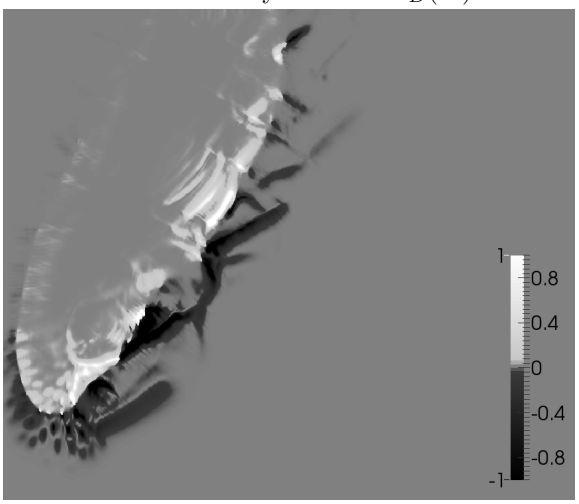
Slice position.



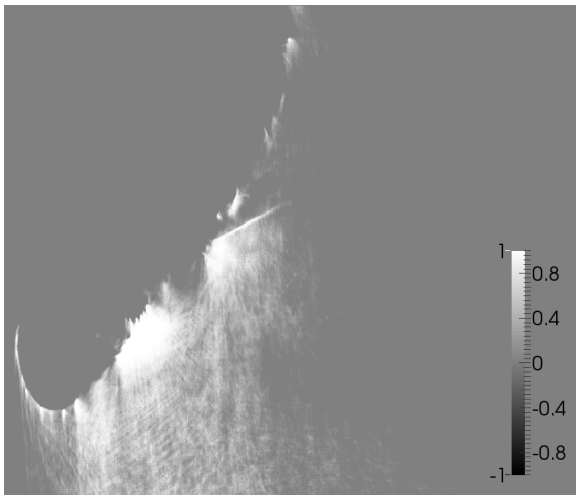
Dense visibility function $w_D(X)$.



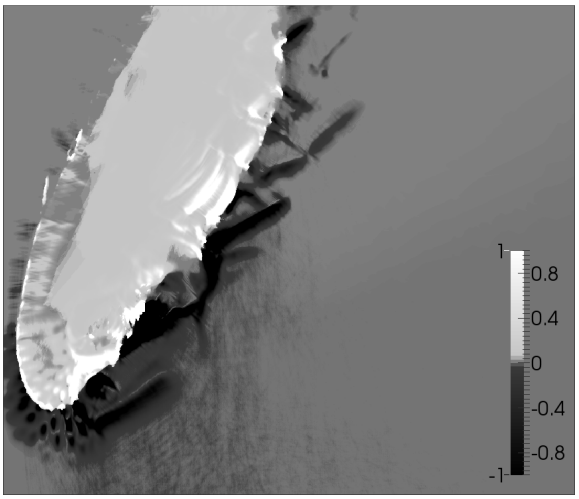
Thin term $f_1(X)$.



Curved term $f_2(X)$.

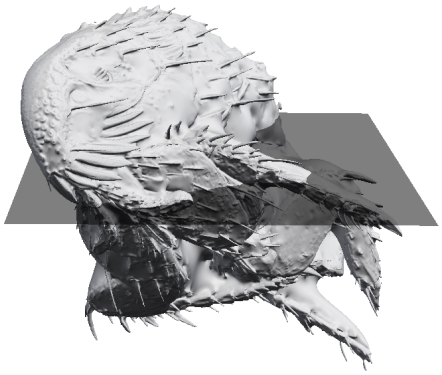


Erosion term $f_3(X)$.

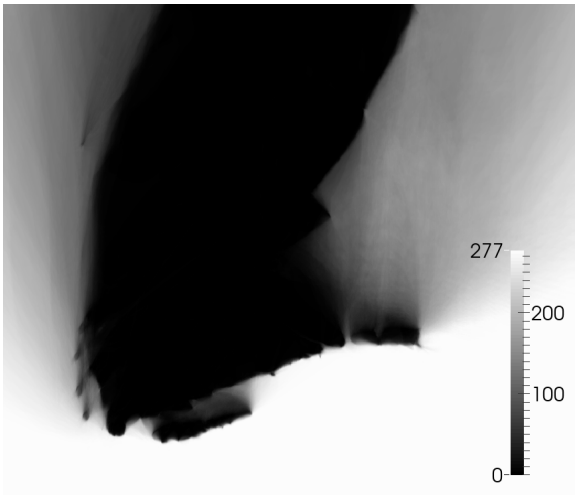


Final regional term $f(X)$.

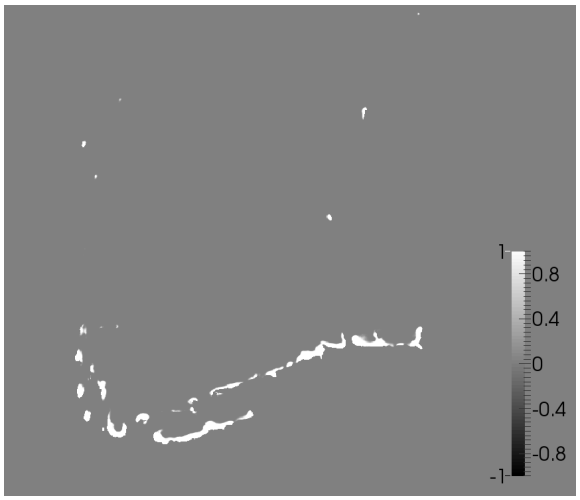
Figure 6.6: Slices through the computed volumetric terms.



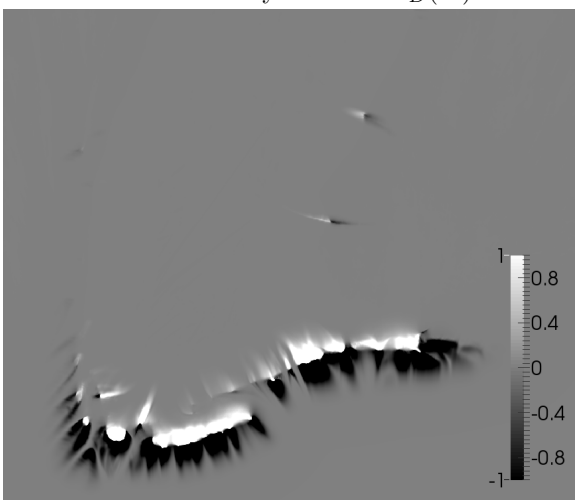
Slice position.



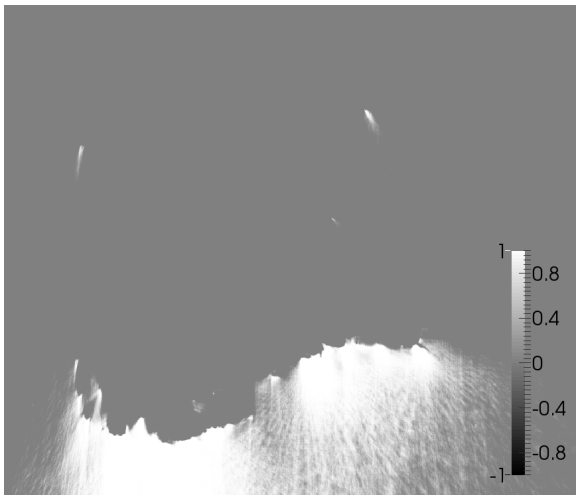
Dense visibility function $w_D(X)$.



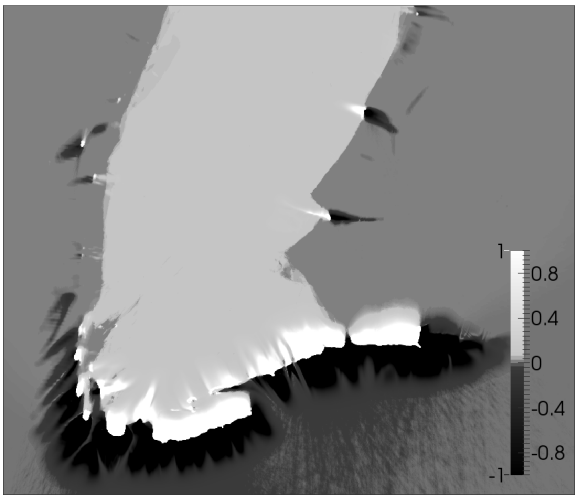
Thin term $f_1(X)$.



Curved term $f_2(X)$.

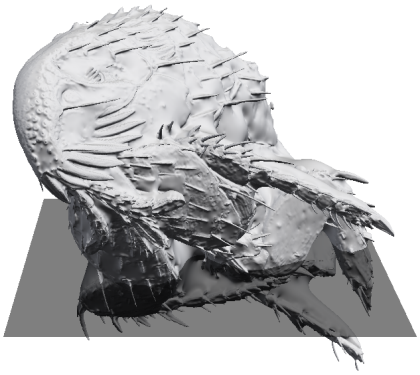


Erosion term $f_3(X)$.

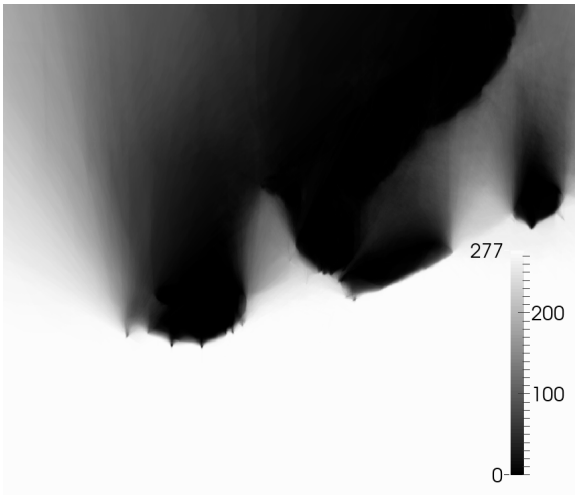


Final regional term $f(X)$.

Figure 6.7: Slices through the computed volumetric terms.

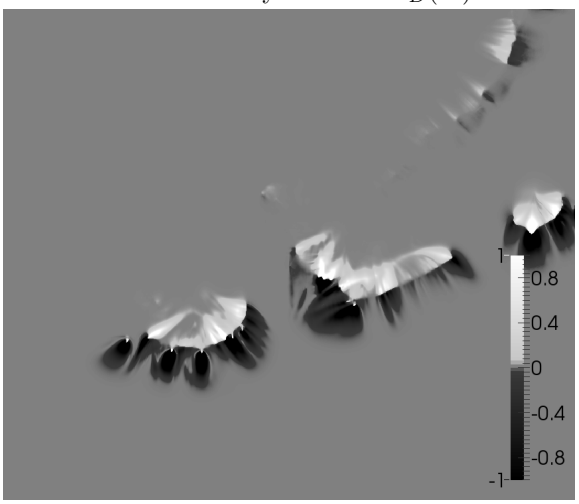


Slice position.

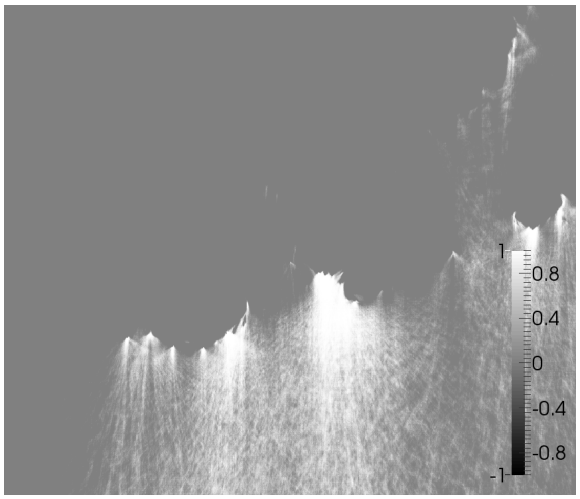


Dense visibility function $w_D(X)$.

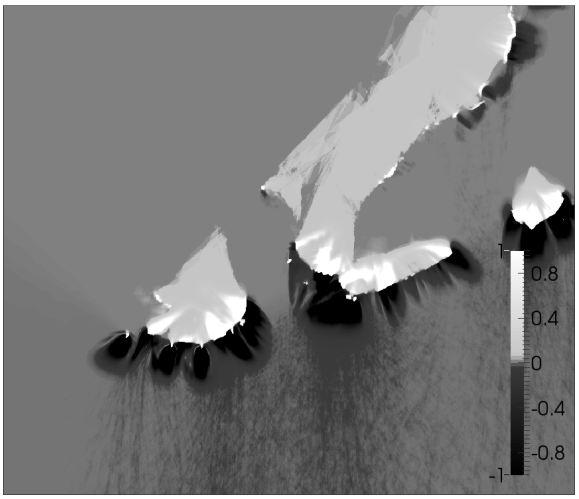
Thin term $f_1(X)$.



Curved term $f_2(X)$.



Erosion term $f_3(X)$.



Final regional term $f(X)$.

Figure 6.8: Slices through the computed volumetric terms.

Figure 6.9: Reconstructed surface. **Left:** the input image. **Right:** the reconstructed shape rendered from the same point of view. Some legs are missing since the reconstruction volume only covers part of the shape.

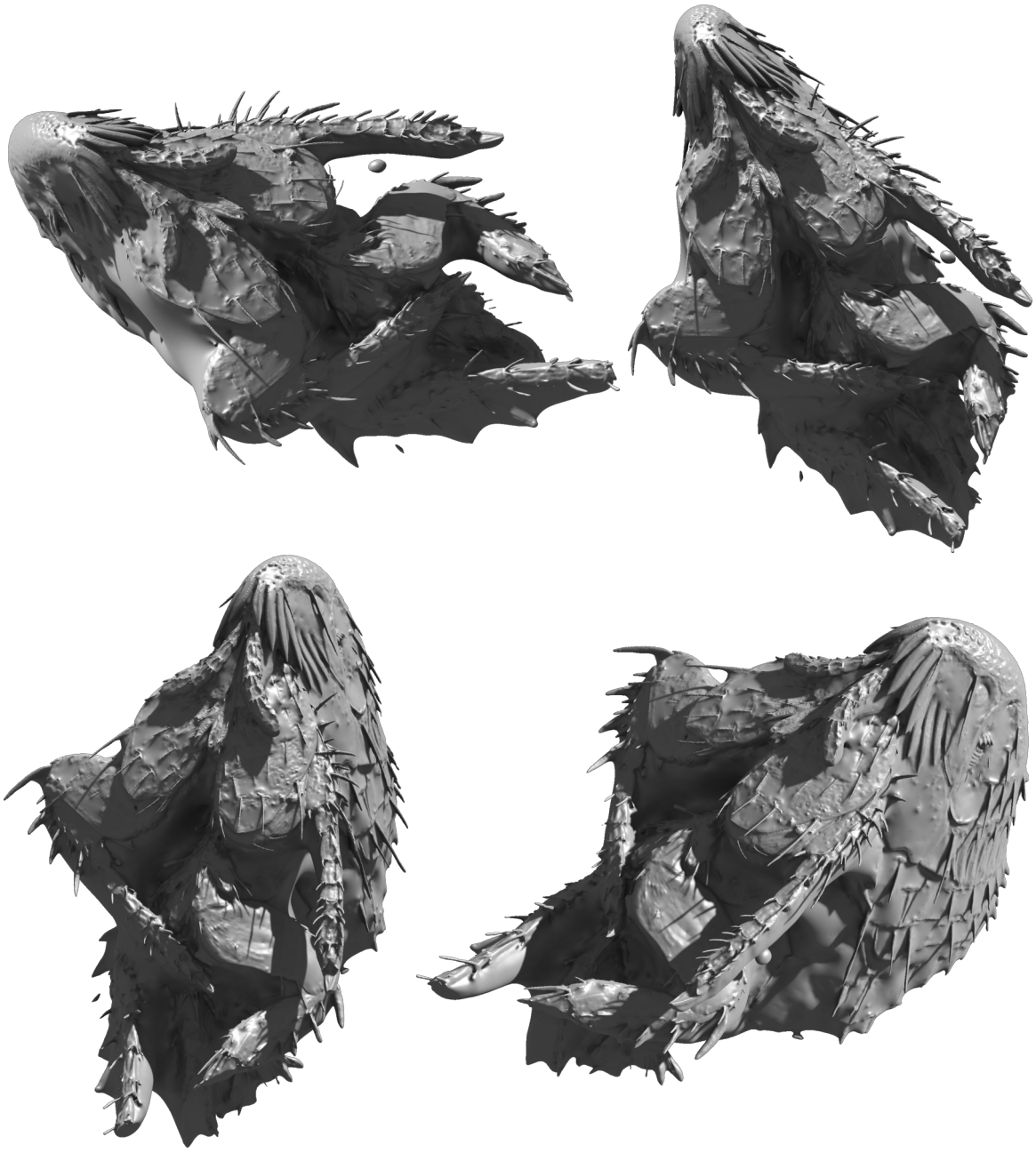


Figure 6.10: Reconstructed surface shown from different sides.

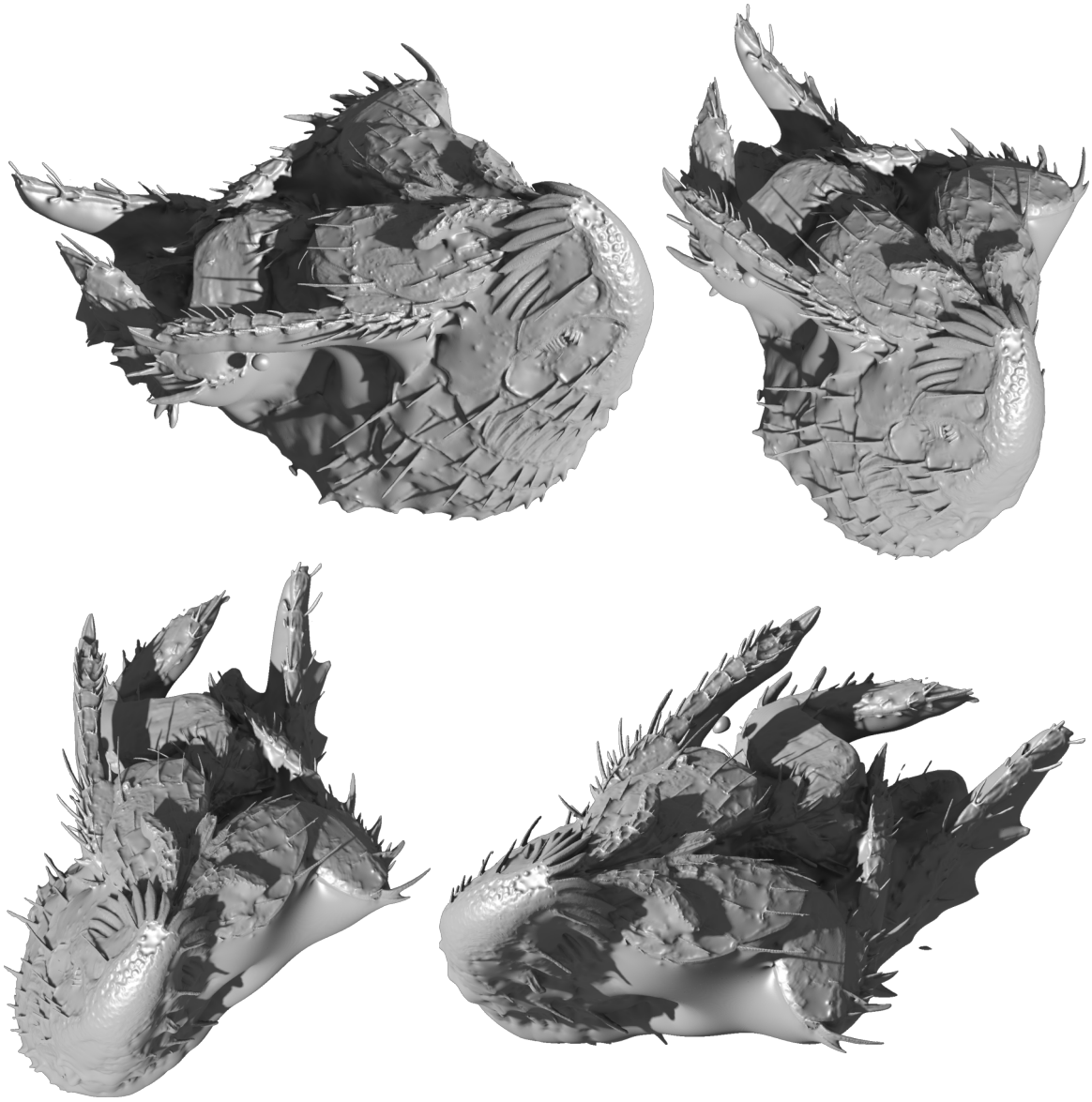


Figure 6.11: Additional views of the reconstructed surface.

6.5 Conclusions

I have presented a method of estimating the shape of an object observed in a medium-baseline grid of images. The method is based on photoconsistency, so it does not only apply to SEM. I conclude that thin features can be reconstructed truthfully by this method as long as they are observed clearly in a sufficient number of images. This is accomplished through a novel anisotropic regularizer and a novel set of regional data terms. Most of the complexity of the method lies in the way in which reliable data terms have to be extracted from the often unreliable dense depth maps. I consider it likely that the surface reconstruction part of the method could be greatly simplified if more reliable depth maps could be obtained.

In the next chapter, we will leave the idea of photoconsistency behind and examine the way in which the appearance of a surface changes as a function the viewing angle. In chapter 8 we will revisit the grid-based surface reconstruction problem and apply that appearance model to obtain a better shape.

Chapter 7

Shading Model for Scanning Electron Microscopy Images

The reconstruction methods presented so far were agnostic to the actual imaging modality. The depths were estimated from an assumption of photoconsistency, while the anisotropic regularizer was derived from observations of image edges. In the following chapters, I will examine the question what the values contained in the pixels actually mean, and how that information can be included in the reconstruction process.

In this chapter, I will propose a novel model of reflectance maps [48] encountered under an electron microscope. In the first part of the chapter, I will define my reflectance model and explain the choices that led to it. In the second part, I will describe a practical method for radiometric calibration, i.e. fitting the model parameters to a given detector setup.

7.1 Model Definition

Every detector attached to the microscope collects electrons of a specific type, and it can only collect the electrons that can reach it. The number of electrons that can reach a detector depends on a number of factors.

1. The beam intensity and energy, the time spent irradiating each pixel and the charge of the detectors. These factors can be controlled, and they only need to be kept constant during calibration and the capture of a set of images.
2. The angle between the beam and the surface determines the total number and angular distribution of emitted electrons. These factors have been studied, and approximate models exist in literature [86]. They are, however, meaningless unless more is known about the detectors.
3. Matter located between the impact point and the detectors acts a shadow caster. This factor depends on the surface being observed, so there is no way of calibrating it.
4. The position and shape of the detectors. These vary among microscopes, so they need to be calibrated for every new detector that is used.

Previous SEM shading models make simplifying assumptions about the reflectance behavior of electrons and the shape and location of the detectors. The latter are commonly assumed to be infinitely small and acting as point lights. My reflectance map makes no such prior

assumptions and is intended to work with arbitrary detector setups. Both the reflectance behavior and the shape of the detector are contained in the same model.

My formulation aims for practical reversibility, i.e. efficient estimation of normals from observed luminance values. It also allows for the combination of an arbitrary number of detectors of arbitrary type.

According to my model, the surface radiance v takes the following form:

$$v \approx a \max(0, p \cdot (n_x, n_y, n_z, 1))^k, \tag{7.1}$$

where $n \in \mathbb{S}^2$ is the surface normal given in eye-space, $a \in \mathbb{R}$ is the local albedo and $p \in \mathbb{R}^4$ and $k \in \mathbb{R}$ are model parameters particular to that specific detector. The reader familiar with computer graphics will recognize the similarity to the reflectance map proposed by Phong [87] and the BRDFs proposed by Blinn [88] and LaFortune [89].

Under $p = (0, 0, 1, 0)$ and $k = -1$, my model generalizes to the inverse cosine model for SE emission that has been applied by Horn [47]. Under $p = (l_x, l_y, l_z, 0)$ and $k = 1$, it is equivalent to a Lambertian reflectance under a directional light $L(\omega) = \delta\omega - l$.

The main purpose of the model is to compute the most likely normal n and albedo a from multiple radiance observations v_i that are made from different viewing directions. If multiple different types of detectors are used simultaneously, then the surface point will exhibit different albedos under the different detectors.

In the case of SE emission, the albedo will primarily describe the soft shadowing. SE images have the appearance of optical images captured under uniform illumination $L(\omega) = \text{const.}$. In computer graphics, this corresponds to an ambient occlusion shading model. There, the degree to which a surface point is shadowed is invariant under rotations of the illumination environment. The shadowing is therefore modeled as a function of position.

In the case of BSE images, the albedo will describe the material composition. According to literature [86], materials that contain heavier atoms will produce a greater BSE yield. My model assumes that the directional distribution of those BSE does not change as a function of material composition. Its effects are therefore modelled by a scalar albedo. Since the shadows cast under BSE detectors are much harder than those under SE, BSE shadows are handled by masking out BSE observations where the observed radiance v_i is too low.

7.2 Radiometric Calibration

The calibration procedure that I propose does not require any specific calibration shapes which could be difficult to obtain or would require a prior estimation of their shape. Instead, it only needs a cylindrical shape such as a length of copper wire. The calibration process is performed along the following steps.

1. Data capture.
2. Estimation of a data-based reflectance map $r_D(\phi, \theta)$.
3. Fit of p and k to $r_D(\phi, \theta)$.

These will be discussed in detail in the following.

7.2.1 Data Capture

The cylindrical shape is placed horizontally under the microscope and recorded at equidistant rotation angles around the vertical axis spanning the full circle. This results in a sequence of images for each detector, since different detectors can be used at the same time. In my experiments I captured 366 such images at 1° angular intervals. This provides an angular overlap of 6 images that is used to correct for small changes in surface appearance that accrue during capture time.

7.2.2 Data-Based Reflectance Map

The captured images show locally rough surfaces, so the data needs to be cleaned first. This is done by computing value histograms along the direction of the cylinder, as follows.

First, the images need to be aligned. I have done this using SE images, but the method should also be applicable to other modalites. An approximate center of rotation $c \in \mathbb{R}^2$ is determined by averaging all images. This average will show two concentric circles corresponding to the two edges of the cylinder. If the center of the rotation is located in the middle of the cylinder, these two circles will coincide.

Next, the angle ϕ_i corresponding to each image i is determined through a hierarchical exhaustive search. This is done by testing a set of initial angles at an initial spacing, and then iteratively repeating the testing for a finer-spaced set of angles centered around the best angle from the previous iteration. A reliable initial angle is given by the capture setup. The quality of an angle ϕ is measured as follows.

Let $q_\phi = (\cos(\phi), \sin(\phi))$ such that it points across the cylinder, i.e. perpendicularly to its edges in the image. A 1D-average array $\hat{v}[q] : \mathbb{N} \mapsto \mathbb{R}$ is then computed as the average of all pixels that project onto the same discrete value of q. The resolution of $\hat{v}[q]$ is chosen to be the image resolution. Then, all pixels near the edge of the cylinder are compared to their corresponding value in $\hat{v}[q]$, and the squared differences are added up, yielding C_ϕ, the cost of the angle. The area near the edge is determined in one initial image and then mapped to the current one using the center of rotation c and the current angle estimate. A bad angle ϕ will lead to a blurry edge in $\hat{v}[q]$ and thus to a large C_ϕ, while only a very precise angle will produce a $\hat{v}[q]$ with an edge that is similarly sharp as the one seen in the image. Because the pixels are projected along the entire length of the image, the theoretical precision limit of this estimation method is equal to $\text{atan}(1/s)$, where s is the resolution of the smaller dimension of the image (usually the height). For a typical 1024×768-image, this corresponds to less than 0.075°.

The directional value average $\hat{v}[q]$ computed for the optimal angle ϕ already contains an estimate of a slice of our reflectance function, but this estimate is not robust to irregularites on the cylinder surface. Particularly in the case of SE reflectance, most local irregularities on the surface lead to excessively bright values, which leads to upward peaks in $\hat{v}[q]$. In order to attain increased robustness against such outliers, we now compute a value histogram analogous to the average.

A 2D-histogram of dimensions $w_H \times h_H$ is allocated, where the width w_H is given by the width of the cylinder in pixels, and the height by the signal resolution of the gray values, i.e. 256 in most cases. Then, all pixels are again mapped onto their corresponding position along the q-axis, and the corresponding value bin is incremented. This histogram is smoothed through a convolution with a Gaussian, and for each q-coordinate, the bin of maximum value along

the vertical axis is selected as the reflectance value r_q for that q. This process is illustrated in Fig. 7.1.

This histogram-based averaging is essentially equivalent to the occlusion-robust denoising proposed in chapter 4. The main difference is the fact that the histogram is smoothed with a Gaussian, while the robust energy in chapter 4 corresponds to a 1D histogram convolved with a truncated parabola. Also, the lateral (q-axis) component of the Gaussian smoothing has no correspondence in the pixelwise independent denoising procedure.

Since we have assumed the surface to be cylindrical, each q corresponds to an inclination angle θ:

$$\theta = \operatorname{asin}((q - q_0)/R), \tag{7.2}$$
$$q_0 = (q_{\max} + q_{\min})/2, \tag{7.3}$$
$$R = (q_{\max} - q_{\min})/2, \tag{7.4}$$

where $q_{\min}$ and $q_{\max}$ are the positions of the cylinder edges along the q-axis. The azimuth angle ϕ is given by the rotation of the image. Together, the reflectance profiles r_q of all images i form a reflectance table $r_{q,\phi}$. The remapping of q to θ is deferred until the next step to avoid additional resampling errors. The different projections that will be discussed in the following are illustrated in Fig. 7.2.

The reflectance table $r_{q,\phi}$ contains redundant information. First, the azimuth angles ϕ at the end of the sequence are repeated. This overlap allows us to compute the ratio $t_{q,\phi}$ of the values observed twice. This ratio is averaged over all q and interpolated linearly over the range that is observed only once. The result is a reflectance map $r'_{q,\phi}$ that wraps smoothly at $\phi = 2\pi$. Further investigation would be needed in order to determine the source of the appearance change which is most prominent in the SE images. Most likely, it is a consequence of the prolonged exposure of the material to the electron beam.

Next, since ϕ covers the full circle while the cylinder is laterally symmetrical, every normal appears in two places in the table. In order to obtain a non-redundant representation, the corresponding values of $r'_{q,\phi}$ are averaged: $r^*(\theta, \phi) \coloneqq 0.5(r'(\theta, \phi) + r'(-\theta, \phi + \pi))$. The values corresponding to $\theta < 0$ (i.e. the left half of $r^*(\theta, \phi)$) are no longer needed, so they are discarded. The indices q have been replaced by the angles θ to increase legibility. The mapping from q to θ is defined in Eq. 7.3.

Finally, the north pole, $\theta = 0$, is repeated at the left edge of each row of $r^*_{\theta,\phi}$ (i.e. the center of each row of $r'_{\theta,\phi}$), because that same frontal normal appears at every azimuth angle ϕ. By extension, normals close to the north pole also take up a larger surface area than normals closer to the equator, which is located at $\theta = \pi/2$. In order to obtain an equal-area representation, each entry $r^*(\theta, \phi)$ is mapped onto its corresponding pixel of a sinusoidal projection $m_{x,y}$,

$$x = \sin(\theta - \pi)\phi \tag{7.5}$$
$$y = \theta. \tag{7.6}$$

The corresponding homogeneous normals $n(\theta, \phi)$ are defined as

$$n(\theta, \phi) = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), -\cos(\theta), 1). \tag{7.7}$$

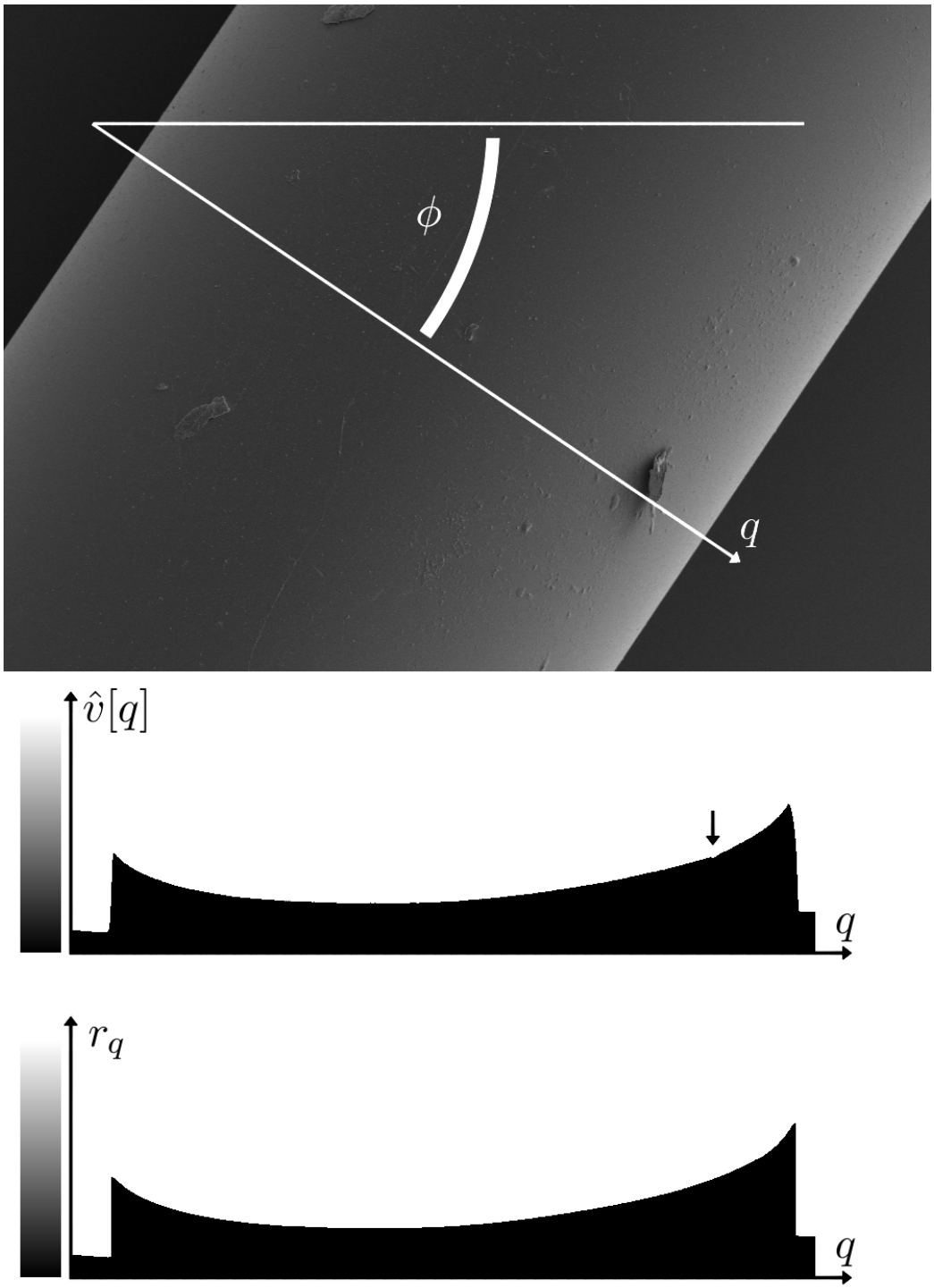


Figure 7.1: An illustration of the radiance estimation process. **Top:** an input image with estimated azimuth angle ϕ. **Center:** the directional value average $\hat{v}[q]$. **Bottom:** The estimated reflectance value r_q. Note the indicated dimple in $\hat{v}[q]$ that is missing in r_q.

7.2.3 Parameter Fit

The two sinusoidal maps thus obtained can now be used to fit the parameters $p \in \mathbb{R}^4$ and $k \in \mathbb{R}$ of the parametric reflectance model. This is accomplished by finding a p and k that

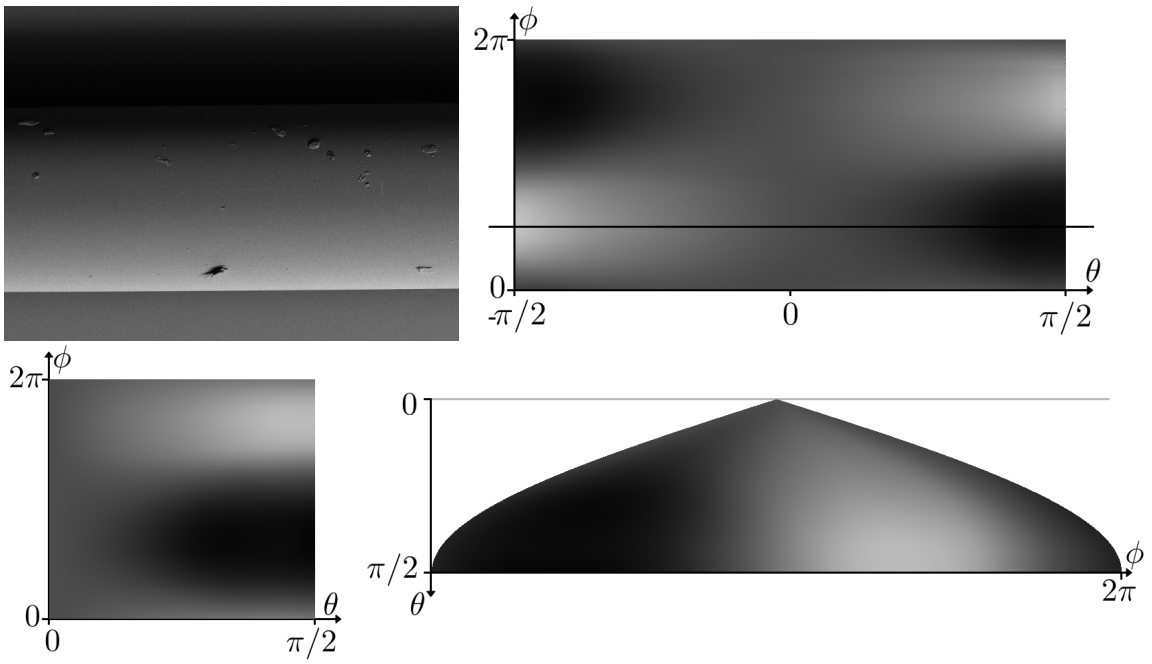


Figure 7.2: The different representations discussed in the text. **Top left:** an input image. **Top right:** the initial redundant reflectance table $r_{q,\phi}$. The horizontal line indicates the position of the displayed image. **Bottom left:** The non-redundant table $r^*(\theta, \phi)$. **Bottom right:** The sinusoidal, area-preserving projection $m_{x,y}$.

minimize

$$C(p,k) \coloneqq \sum_{x,y} \Big(m_{x,y} - \max(0, p \cdot n_{x,y})^k \Big)^2. \tag{7.8}$$

Since C is smooth, this can be done using the Nelder-Mead downhill simplex algorithm [90]. Because C is not convex for $|k| < 1$, the fitting procedure is repeated for both signs of k as initial conditions.

7.3 Experiments

I have performed above procedure for two detectors attached to an FEI Versa SEM. First, an Everhart-Thornley SE detector (ETD) that is located in front and slightly to the right of the scene from the point of view of the image. The grid voltage of the detector was set to zero. Second, the lower 120° segment of an annular (i.e. ring-shaped) BSE detector (ABS) mounted around the objective lens.

In both cases, the optimal fit is obtained by a positive k. The optimal parameter values are as follows:

$$\begin{aligned} \textbf{ABS:} \quad & p = (0.1616, 0.3225, -0.0276, 0.4565) \\ & k = 1.6470 \\ \textbf{ETD:} \quad & p = (0.1240, -0.0471, 0.3269, 0.7423) \\ & k = 1.5933 \end{aligned}$$

Visualizations of the corresponding reflectance maps are shown on the following two pages.

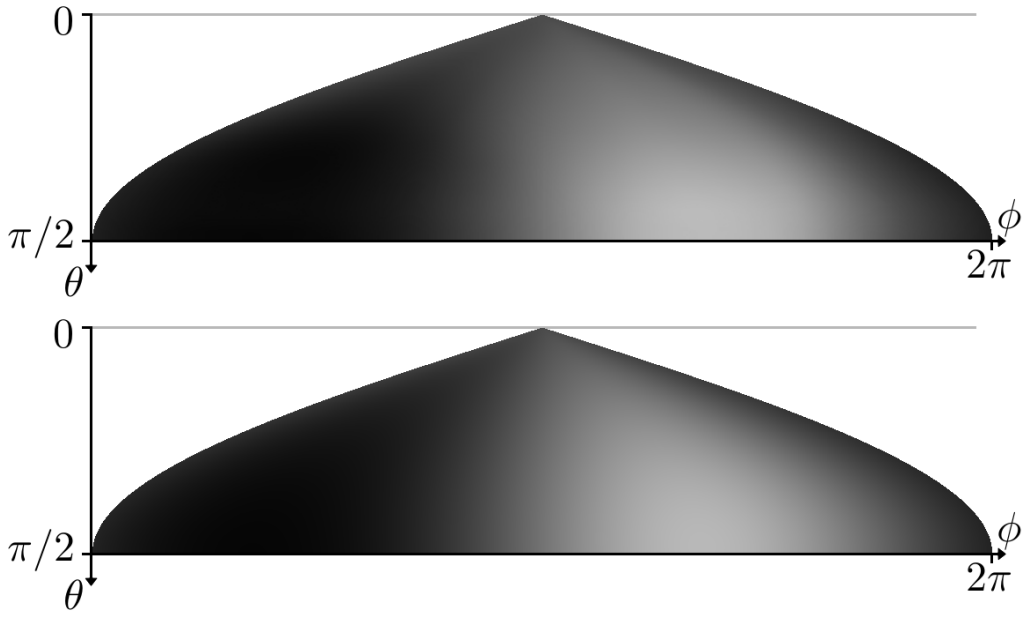


Figure 7.3: The reflectance of the lower 120° segment of an annular BSE detector (ABS). **Top:** the measured reflectance map. **Bottom:** the approximation by my model.

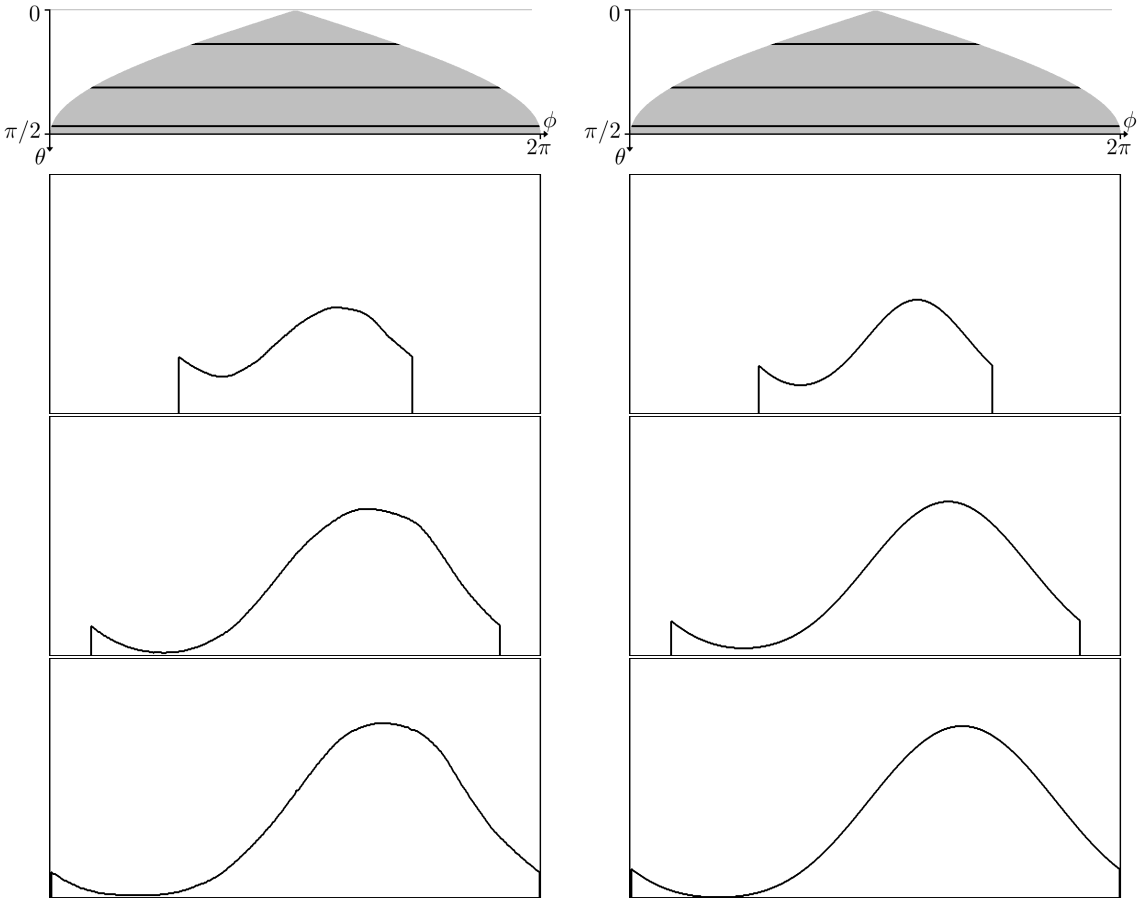


Figure 7.4: Graphs of the reflectance along the indicated θ values. **Left:** measured data. **Right:** model fit.

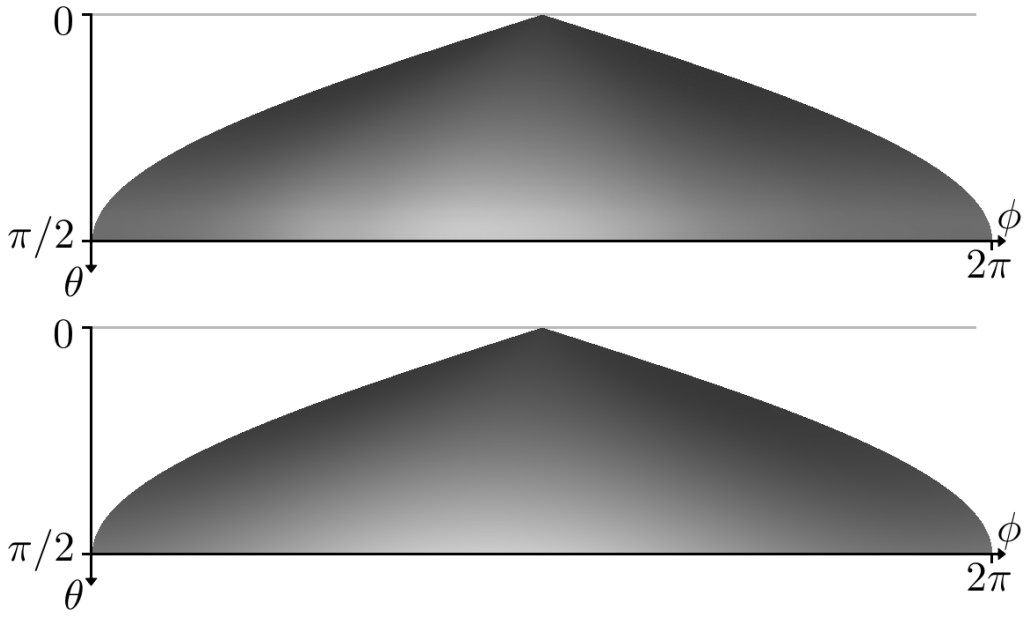


Figure 7.5: The reflectance of an Everhart-Thornley SE detector at zero potential (ETD). **Top:** the measured reflectance map. **Bottom:** the approximation by my model.

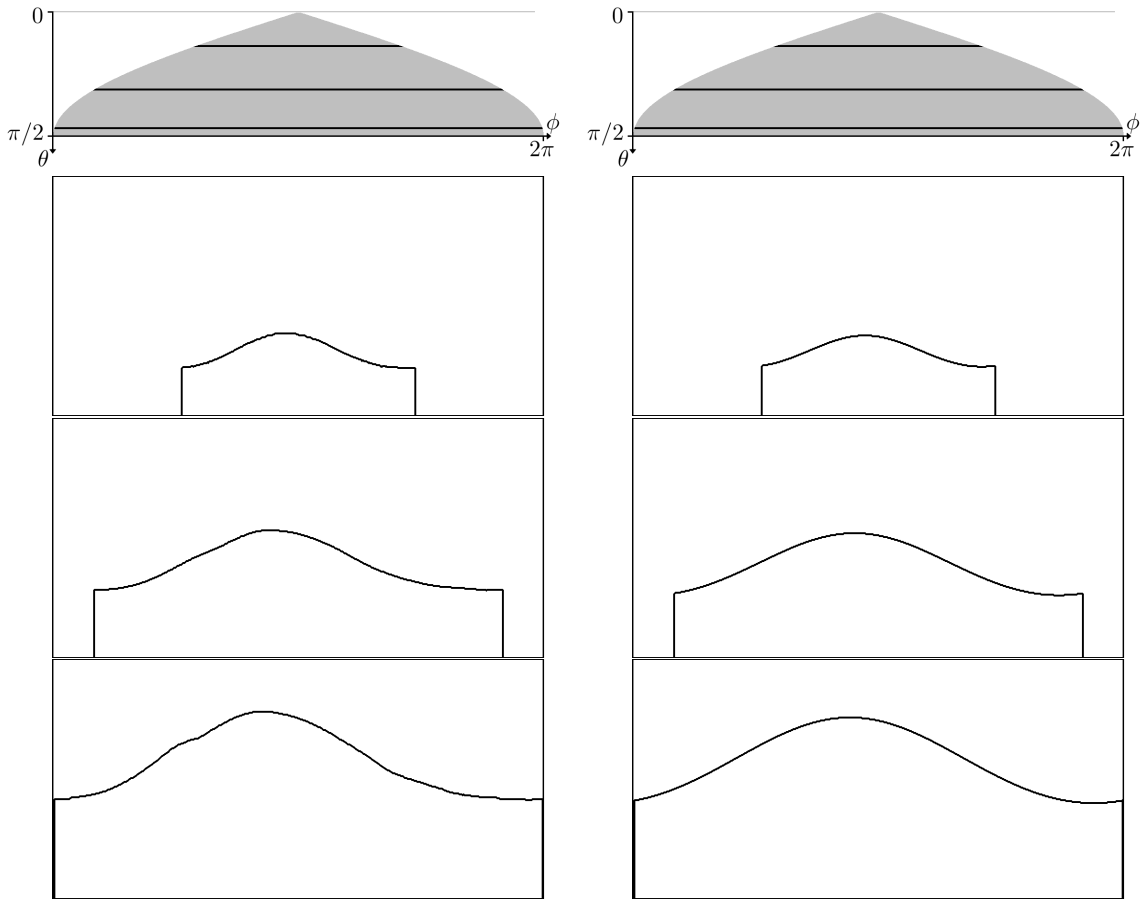


Figure 7.6: Graphs of the reflectance along the indicated θ values. **Left:** measured data. **Right:** model fit.

7.4 Conclusions

I have presented a novel empirical shading model for SEM images and proposed a means of measuring its parameters. Although the model does not fit the measured reflectance tables perfectly, the approximation appears very similar under cursory visual inspection. In the next chapter, I will describe how the specific formulation of the model can be used to efficiently infer the surface normal from a set of observations.

Chapter 8

Normal Estimation from Shading

Now that we have established a practical shading model, I will show how it can be used to infer the surface normal and albedos from a set of observations of a point under different viewing angles and detectors. Recall that the albedo is different under each detector. The justification for this was provided at the end of 7.1.

In this chapter, I will present an algorithm that combines image depths obtained through MVS with surface normals inferred from the shading to reconstruct the entire surface. This is similar in spirit to the algorithms by Ikeuchi [56, 57] and Beil and Carlsen [69] for optical and SEM images, respectively. The details of the algorithm differ in a number of respects, however. For one, the MVS part of my algorithm — which was already presented in 6.1 — does not follow a coarse-to-fine scheme, allowing it to estimate the depths of very fine features that would disappear at coarser scales. More importantly, the normal estimation is performed using my general shading model formulation, while that of Beil and Carlsen relies on a symmetrical two-detector arrangement. In addition, my algorithm works on an arbitrary number of images, while theirs has been developed for exactly two views.

The algorithm described in this chapter is outlined as follows. First, the sparse and dense depths are estimated as described in 6.1. Then, the surface normals are estimated at the positions implied by the dense depth maps, yielding normal maps corresponding to those depth maps. Those normals are then used to integrate new, *curved* depth maps that conform to the reliable depth values in the sparse depth maps. From the curved depths, regional terms are computed as described in 6.3, and a watertight surface is reconstructed. Since the actual normals are now known for each pixel, the anisotropic regularization tensor D is constructed from those, instead of the edges.

In chapter 6, we used a TV regularizer for surface reconstruction. The advantage of the TV regularizer was its ability to preserve sharp boundaries of regions. Since we now know where the sharp edges are located, as they are visible in the shading and thus in the normal maps, we now use an L^2 regularizer instead. Unlike a TV regularizer, the L^2 variant offers a stronger guarantee that the areas away from those edges will indeed be smooth. The new reconstruction problem is formulated as follows:

$$E_{L^2}(u) \coloneqq \int_\Omega m(X)(f_C(X) - u(X))^2 + |D_N(X)\nabla u(X)|^2 dX, \tag{8.1}$$

where $f_C : \mathbb{R}^3 \mapsto \{-1, 0, 1\}$ is the regional term derived from the curved depth maps, $m : \mathbb{R} \mapsto [0, 1]$ is its weight and D_N is the regularization tensor constructed from the estimated normals. Note that the energy E_{L^2}, unlike energy E_{TV} from Eq. 6.1, does not

contain a product between f and u. For this reason, there is no longer a need to constrain the value range of u to the unit interval $[0, 1]$. The optimal u is found using biased anisotropic diffusion analogous to Eq. 6.28 on page 68.

In the following, I will describe the individual steps of the algorithm. We assume that sparse and dense depth maps z^* and $z^{(D)}$ have been computed as outlined in 6.1. Next, we proceed to estimate the normals n at the points on the surface of $z^{(D)}$.

8.1 Normal Inference

Recall the observation we made in chapter 4, that estimating a denoised value does not require a correct depth, because in a smooth area, different depths will lead to similar gray values and thus to a similar denoised value. In the following procedure, we will apply this observation to surface normals. The depths in the interpolated, dense depth maps $z^{(D)}$ are reliable near the observed edges, but they suffer from a planarity bias (i.e. type 2 degradations) in the smooth areas. Although the depths in those areas are not precise, I assume that they will still produce similar gray values as the correct depths, and thus allow us to estimate a reliable surface normal.

Every pixel q in the dense depth map $z^{(D)}$ can be back-projected to its corresponding world-space position Q_w. That position is then projected into all neighboring views i, yielding an image-space position q_i. The gray values $v_{i,d}$ seen at q_i in the images captured by all detectors d are extracted. Furthermore, the dense depth map of i is used to determine the visibility of q_w in that image. To that end, the depth of the projected point is compared to the corresponding depth in $z_i^{(D)}$, the dense depth map of image i. If the point p_w is located behind the surface in i, then we set its occlusion depth δ_i to the difference between the two depths. Otherwise, δ_i is set to zero.

Now, the surface normal n can be estimated from the observations $v_{i,d}$ and occlusion depths δ_i. Although we are only interested in the normal itself, the albedos a_d of all considered detectors d also need to be estimated because they are part of the appearance model.

The estimation of the most likely parameters a and n is performed using an alternating least-squares scheme. First, the normal is estimated as a free $\mathbb{R}^3$-vector n_F under an assumed albedo $a = 1$. The resulting vector is normalized: $n \coloneqq n_F/|n_F|$. Then, the albedos a_d for all the detectors D are estimated assuming n as the surface normal. The process is then repeated, using the resulting albedos a_D. In my experiments, this led to very quick convergence, so only few iterations were needed.

Formally, the parameters are estimated as follows. Let p_d and k_d be the shading parameters of detector d. Then, we are looking for a minimum,

$$n_F \coloneqq \operatorname*{argmin}_{n\in\mathbb{R}^3} \sum_{i,d} w_{i,d}\Big(v_{i,d} - a_d \max(0, p_d \cdot \bar{V}_i(n_x, n_y, n_z, 1))^{k_d}\Big)^2, \tag{8.2}$$

where $\bar{V}_i$ is the 4×4 view matrix of view i with its translational components removed, i.e. $\bar{V}_{i,1,1} = \bar{V}_{i,1,2} = \bar{V}_{i,1,3} = 0$ and $\bar{V}_{i,1,4} = 1$ and the weights $w_{i,d}$ consist of the shadow weights $w_{i,d}^{(s)}$ and the occlusion weights $w_{i,d}^{(o)}$,

$$w_{i,d} \coloneqq w_{i,d}^{(s)} w_{i,d}^{(o)}. \tag{8.3}$$

The occlusion weights serve to exclude views from which the point is occluded, and are defined as

$$w_{i,d}^{(o)} \coloneqq \exp(-\delta_i/\tau_\delta). \tag{8.4}$$

The shadow-weights are equal to one for SE images, and for BSE images, they are given by

$$w_{i,d}^{(s)} \coloneqq 1 - \exp(-v_{i,d}^2/\tau_s^2). \tag{8.5}$$

This treatment of shadows has been motivated at the beginning of the previous chapter.

The function in Eq. 8.2 could be minimized using an iterative method, but that would be very expensive, since we are dealing with millions of pixels in each of the hundreds of images. Instead, the equation is slightly simplified to allow for a closed-form solution,

$$n_F \approx n_F^* = \underset{n\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{i,d} \widetilde{w}_{i,d} w_{i,d} \Big(v_{i,d}^{1/k_d} - a_d^{1/k_d} p_d \cdot \bar{V}_i (n_x, n_y, n_z, 1) \Big)^2. \tag{8.6}$$

We have introduced two alterations. First, the k-th power has been moved to the left hand side by transforming both sides by $h(x) : x \mapsto x^{1/k_d}$. To account for the reweighting of the errors due to this nonlinear transformation, new weights $\widetilde{w}_{i,d}$ are introduced. These are given by the squared inverse of the derivative of $h(x)$ taken at the observed gray value,

$$\widetilde{w}_{i,d} \coloneqq (h'(v_{i,d}))^{-2} = k_d^2 v_{i,d}^{2(k_d-1)/k_d}. \tag{8.7}$$

Second, the max operation has been removed. To account for this, observations with a negative $(p_d \cdot \bar{V}_i(n_x, n_y, n_z, 1))$ are discarded in the next iteration. Similarly, observations from views in which the estimated normal n points away from the camera are also discarded.

The simplified function in Eq. 8.6 can be solved linearly,

$$n_F^* = (A^t A)^{-1} (A^t b), \tag{8.8}$$

where A is an $|i,d| \times 3$ matrix, each row of which corresponds to an (i,d) combination and is given by,

$$A_{i,d} \coloneqq \widetilde{w}_{i,d} w_{i,d} a_d^{1/k_d} (\bar{V}_i^t p_d)_{1,2,3}^t. \tag{8.9}$$

The notation $v_{1,2,3}$ refers to the first three components of v. The evidence vector b is given by,

$$b_{i,d} \coloneqq \widetilde{w}_{i,d} w_{i,d} (v_{i,d} - a_d^{1/k_d} p_{d,4}). \tag{8.10}$$

Once a normal estimate n is available, the estimation of the corresponding albedos can be accomplished through linear regression. Let n_H be the homogeneous normal, i.e. $n_H \coloneqq (n_x, n_y, n_z, 1)^t$. Then,

$$a_d = \left(\underset{a\in\mathbb{R}}{\operatorname{argmin}} \sum_i \widetilde{w}_{i,d} w_{i,d} \Big(v_{i,d}^{1/k_d} - a(p_d \cdot \bar{V}_i n_H) \Big)^2 \right)^{k_d} \tag{8.11}$$

$$= \Big(\frac{\sum_i \widetilde{w}_{i,d} w_{i,d} (p_d \cdot \bar{V}_i n_H) v_{i,d}^{1/k_d}}{\sum_i \widetilde{w}_{i,d} w_{i,d} (p_d \cdot \bar{V}_i n_H)^2} \Big)^{k_d} \tag{8.12}$$

After the normal inference procedure is complete for each pixel of image i, we have access to a normal map $n_i(x,y)$. In the following section, I will describe how these normals are used to obtain a better depth map $z^{(C)}$ that does not suffer from the same fronto-planarity bias as the interpolated depth $z^{(D)}$.

8.2 Normal Integration

The estimation of the curved depth map $z^{(C)}$ is analogous to the estimation of $z^{(D)}$ described in 5.1. Formally, it corresponds to finding a depth map that minimizes,

$$E_{\mathrm{C}}(z^{(C)}) \coloneqq \sum_p \Big(c_p (z_p^{(C)} - z_p^*)^2 + \mu_C \sum_{q \in N(p)} a_{pq} (z_p^{(C)} - z_q^{(C)} + s_{pq})^2 \Big). \tag{8.13}$$

The only difference to Eq. 5.1 are the slope coefficients s_{pq} that indicate the surface orientation between pixels p and q. Formally, they are given by

$$s_{pq} = \frac{\bar{n}_{pq}^t K_i^{-1} d_{pq}}{\bar{n}_{pq}} z_{pq}^{(C)}, \tag{8.14}$$

where $\bar{n}_{pq}$ is the average eye-space normal between p and q,

$$\bar{n}_{pq} \coloneqq R_i \frac{m}{|m|} \tag{8.15}$$

$$m \coloneqq n_p + n_q, \tag{8.16}$$

R_i is the rotational part of the view matrix V_i, K_i is the camera matrix, d_{pq} is the image-space vector from p to q, $(q_x - p_x, q_y - p_y)^t$ and $z_{pq}^{(C)}$ is the average depth of p and q. In practice, $z_{pq}^{(C)}$ is kept constant at $z_{pq}^{(C)} = V_{i3,4}$ across the entire scene, since the perspective foreshortening effect under an electron microscope is always very small. This allows us to determine the slopes s_{pq} once in the beginning and to keep them constant afterwards.

The pairwise pixel-affinities a_{pq} are slightly changed from those in Eq. 6.4 (on page 60) to also incorporate differences in the normal between the pixels p and q:

$$a_{pq} \coloneqq \exp\Big(-\frac{1}{\lambda_S^2}(v_{S,p} - v_{S,q})^2 - \frac{1}{\lambda_N^2}(n_p - n_q)^2 \Big). \tag{8.17}$$

Since the normals n are computed from multiple gray values, they are less noisy than the gray values v_S and v_B themselves. In turn, the back-scattered gray values v_B are no longer used, since they contain cast shadows that can introduce unwanted edges into the depth map.

The optimization of Eq. 8.13 is performed analogously to that of Eq. 5.1, using the procedure defined in Eq. 5.3. I have found the hierarchical approach to be too unstable in this case. For this reason, the optimization is performed at the highest resolution level directly.

The resulting depth maps exhibit fewer degradations of type 2, so smoothly curved surfaces are better preserved.

8.3 Surface Reconstruction

The surface reconstruction process is similar to that described in chapter 6, except for the regional terms f_C and m that replace the term f, the regularizer D_N that is constructed from the normals and the cost function itself. The cost function has already been defined in Eq. 8.1.

In the following, I will define and describe the individual terms m, f_C and D_N.

8.3.1 Regional Terms

The new regional terms are constructed by first computing the dense outside field w_C from the new, curved depth maps $z^{(C)}$ analogous to the computation of w_D from $z^{(D)}$ in 6.3.1. From w_C, I compute the regional term f as described in Eq. 6.16. Then, f_C and m are given by,

$$f_C(X) = \begin{cases} -1 & \text{if } f(X) < -\epsilon \\ 1 & \text{if } f(X) > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{8.18}$$

$$m(X) = f_C(X)^2. \tag{8.19}$$

8.3.2 Regularization Tensor

The regularization tensor D_N is still computed from a pseudo structure-tensor J_N, but J_N is itself determined from the surface normals, and not the edges. It is defined as follows:

$$J_N(X) \coloneqq G_{\rho_N} * \sum_i \exp(-\frac{\bar{\delta}_i^2}{\sigma_D^2} n^t(X) n(X)), \tag{8.20}$$

where $\bar{\delta}_i$ is the distance between point X and the corresponding surface point in $z^{(C)}$, the curved depth map of image i, and n_i is the surface normal at that point. The normal is assumed to be given as a column vector, so $n^t n$ is a dyadic product, i.e. a 3×3 rank 1 matrix. $G_{\rho_N} *$ denotes a convolution with a Gaussian.

The construction of D_N begins by computing $\widetilde{D}_N$ from J_N according to Eq. 6.15. Then, D_N is attenuated in areas where many points are observed,

$$D_N(X) \coloneqq \frac{1}{c_s(X)} \widetilde{D}_N(X), \tag{8.21}$$

where c_s is the confidence-weighted point density as defined in 6.3.3. This way, areas where few points are observed are reconstructed mostly from the normals, while in the areas around the observed edges, where many point observations are available, the reconstruction is based on photoconsistency.

8.4 Experiments

The proposed procedure was applied to the images described and the depth maps computed in chapter 6. A subgrid of 5×7 images around the reference view (comprised of 5 tilt angles and 7 rotation angles) have been considered for the estimation of surface normals. These normals have then been used to integrate curved depth maps. Comparisons between the interpolated and the curved depth maps are shown in Figs. 8.2 and 8.3. A comparison of the two resulting outside fields $w_D(X)$ is shown in Fig. 8.1. Recall that these fields indicate the number of depth maps that see a given point X in front of their respective surface. Renderings of the final shape are displayed in Figs. 8.4 and 8.5.

The following parameter values were used:

Normal estimation:	$\tau_\delta = 0.01mm$	$\tau_\sigma = 0.1$		
Depth integration:	$\mu_C = 30$	$\lambda_S = 0.02$	$\lambda_N = 0.2$	
Regularizer:	$\sigma_D = 0.003mm$	$\tau_V = 3$	$k_C = 2$	$k_P = 15$
Regional terms:	$\alpha_c = 0.01$	$\alpha_1 = 0.5$	$\alpha_2 = 0.5$	$\alpha_3 = 0.3$
Certain term:	$\tau_{\text{in}} = 5$	$\sigma_{\text{in}} = 2$	$\tau_{\text{out}} = 80$	$\sigma_{\text{out}} = 200$
Curved term:	$\mu_{\text{GVF}} = 500$			
Erosion term:	$\tau_3 = 0.05N_i$	$\sigma_3 = 0.1N_i$		

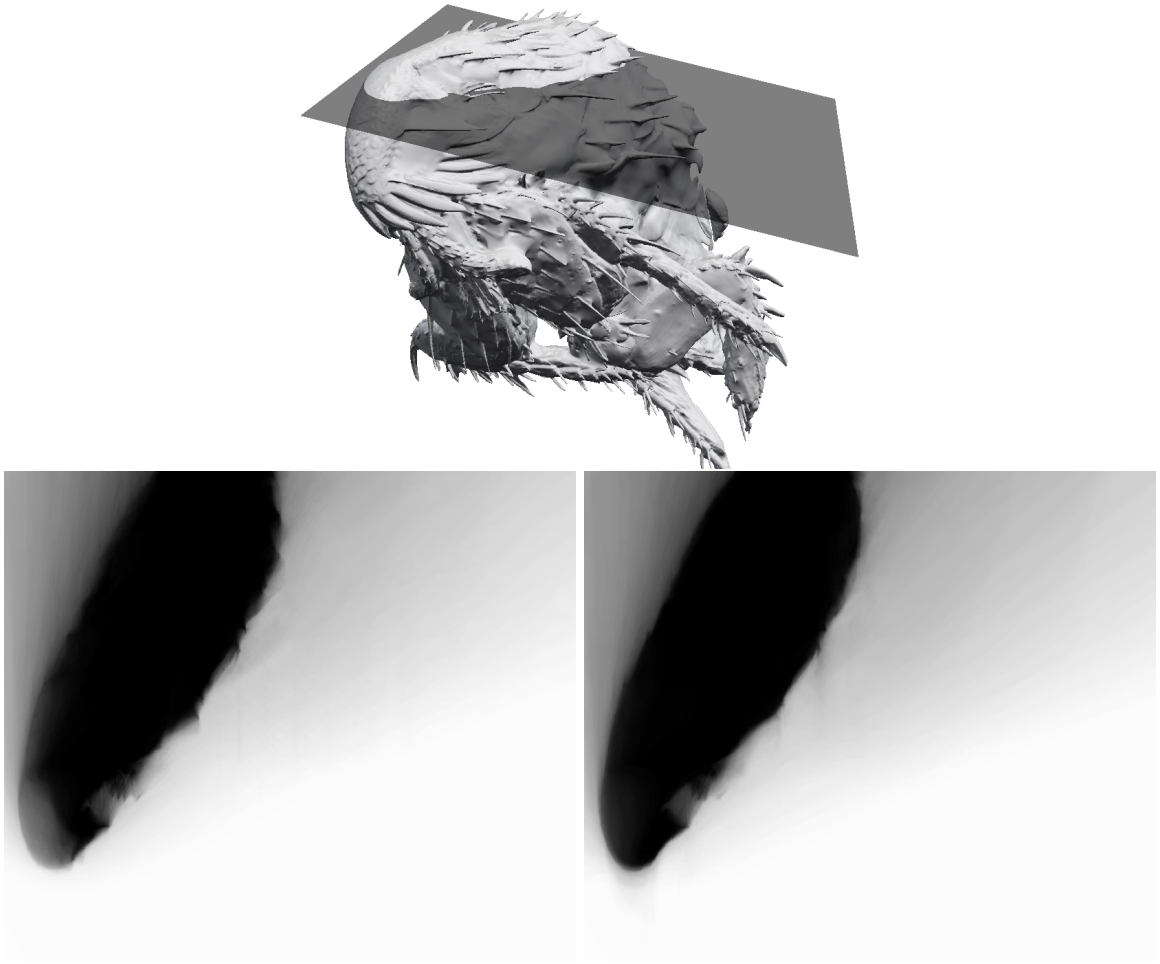


Figure 8.1: Comparisons of the dense outside fields w_D and w_C resulting from the flat and the curved depth maps respectively. **Top:** position of the slice. **Bottom left:** flat outside field w_D. **Bottom right:** curved outside field w_C. Note the decrease in type 2 destructive degradations, i.e. the smaller number of views that erroneously see the inside of the head as outside. The reason for this are the improvements in the depth maps as shown in Fig. 8.3.

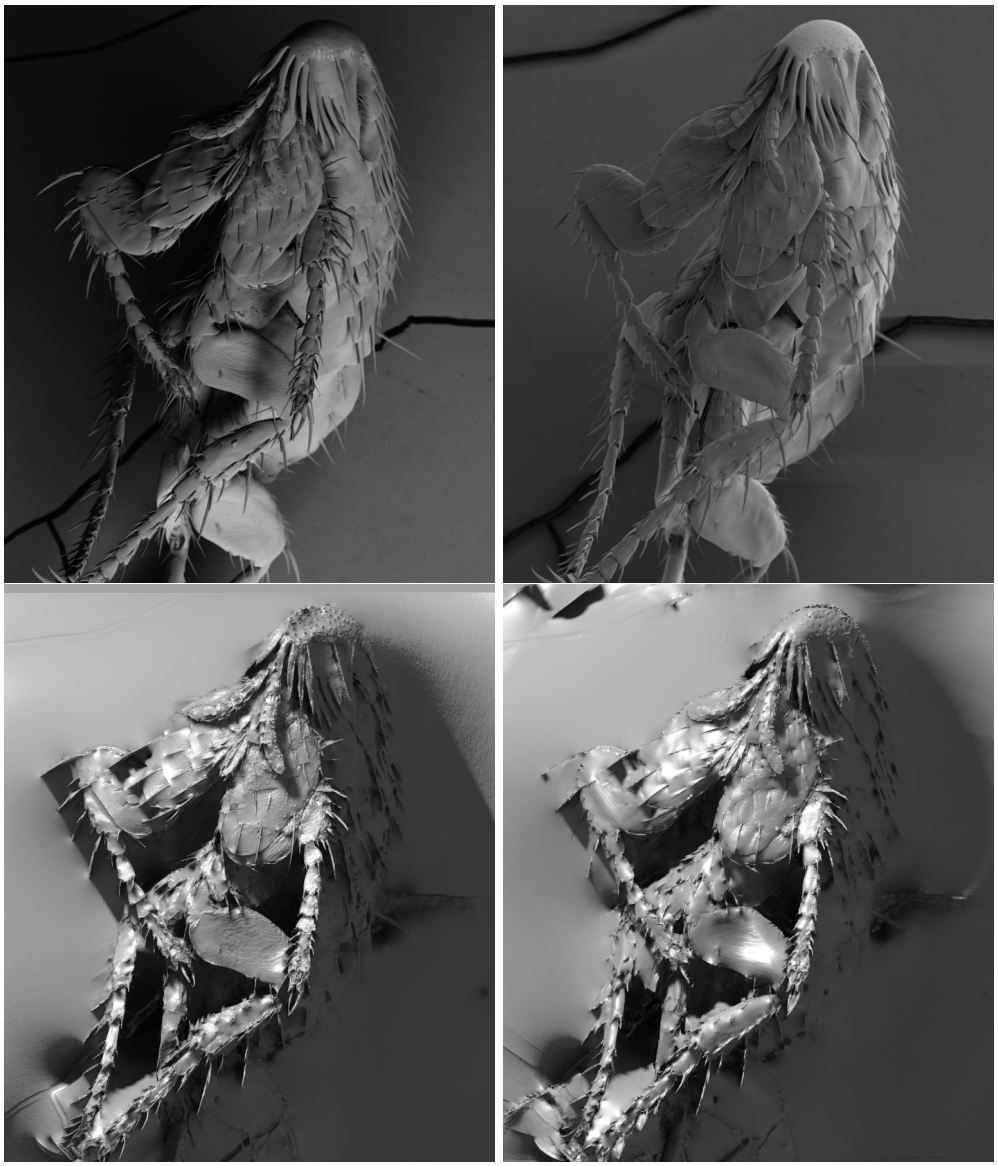


Figure 8.2: Comparison of the flat, interpolated depth maps $z^{(D)}$ and the curved, integrated ones, $z^{(C)}$. **Top left:** the BSE image corresponding to the reference view. **Top right:** the corresponding SE image. **Bottom left:** the flat depth map $z^{(D)}$. **Bottom right:** the curved depth map $z^{(C)}$. The bottom images have been rendered under an arbitrary illumination to showcase the differences in surface orientation. All images are shown from the viewing direction given by the reference view.

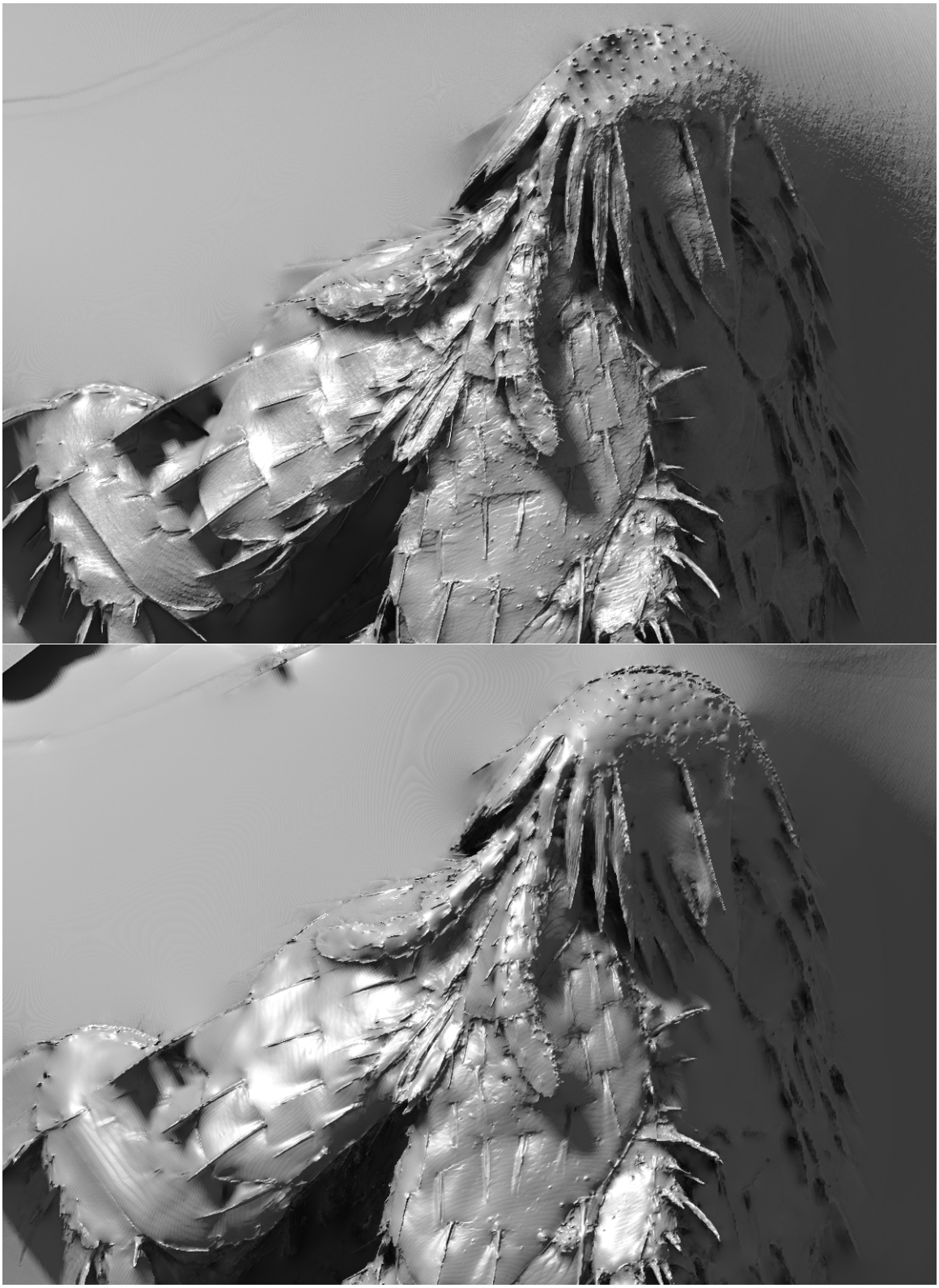


Figure 8.3: Enlarged versions of the renderings from Fig 8.2. Note the depression in the head of the animal in $z^{(D)}$ (top) that is removed by the integration along the estimated normals (bottom).

Figure 8.4: Reconstructed surface. **Left:** photoconsistency-based reconstruction from chapter 6. **Right:** shading-based reconstruction using the method presented in this chapter. Note the improvements in the smooth areas and the reduction in sheets forming outside the object. The latter is a consequence of the better outside field w_C which allows us to use a smaller tolerance when classifying points as outside.

Figure 8.5: Reconstructed surface shown from different sides. The views correspond to those shown in Fig. 6.10 on page 78

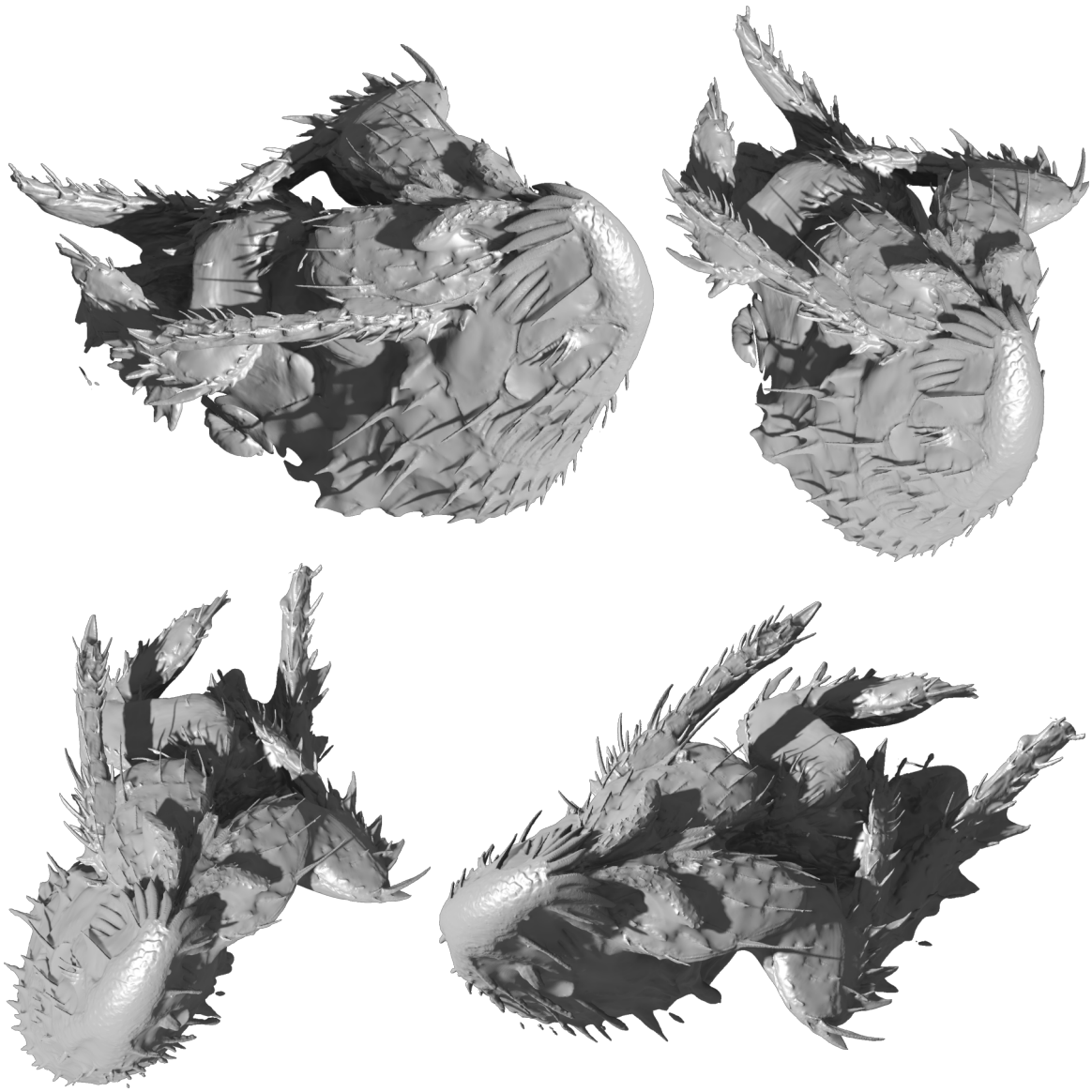


Figure 8.6: Additional views of the reconstructed surface. The views correspond to those shown in Fig. 6.11 on page 79

8.5 Conclusions

I have presented a method that augments the reconstruction procedure presented in chapter 6 through surface normal estimates obtained from the observed shading. As can be seen in the figures, this leads to visible improvements of the reconstructed shape. First, smoothly curved areas are reconstructed more faithfully and show fewer spurious edges. Second, the more reliable dense depth maps allow for a stricter classification of points observed in front of those depth maps (i.e. points where w_C is greater than zero) as outside. This in turn suppresses the formation of spurious sheets that would otherwise form in empty space between features (see Fig. 8.4). The method is, however, ultimately limited by unreliable sparse depth maps that are computed under the assumption of photoconsistency. In the next chapter, I will propose a depth estimation method that avoids the concept of photoconsistency, and that aims to estimate the pixelwise depths themselves using the shading model.

Chapter 9

Depth Estimation from Shading

The reconstruction method presented in the previous chapter exploits the complementarity between MVS and PS. While MVS allows us to estimate the depth around sharp edges, PS allows us to interpolate that depth across the smooth areas in-between. This still leaves two conceptual problems open: first, the depth estimation can only be performed from a small subset of views, since the surface would undergo an excessive radiance change otherwise. Second, not all features exhibit sufficiently sharp edges on their surface.

In this chapter, I will discuss the prospect of estimating the depth itself using the shading model. In place of photoconsistency, we will consider the concept of shading-consistency. This means that instead of assuming that the point will appear with the same gray value in the images, and then working around the fact that it does not, we will instead look for a depth at which the same set of shading parameters n and p explains the maximal number of observations.

In the first part of the chapter, I will propose a hypothetical algorithm that extends the denoising depth estimation method from chapter 4. Such an extended algorithm is not feasible using the computing machines available at the time of this writing. In the second part, I will describe a less general algorithm that allows us to estimate the depth from the shading today, although it relies on a specific capture setup.

9.1 Motivation

My previous algorithm uses depth maps obtained through fronto-planar interpolation (which were first introduced in chapter 5) to determine the points in space at which normals are estimated. We have made the following explicit assumption: if the interpolation dislocates a surface point in a smooth area by δz, then that new point will lead to similar gray-value observations and thus to a similar normal estimate. Although this assumption certainly holds for small δz, we do not have any guarantee that the fronto-planar interpolation will not dislocate the point excessively.

As a thought experiment, imagine a perfectly smooth sphere captured by an electron microscope. The only edge in that image will be a circular outline. Using the concept of a transient depth established in chapter 5, it is indeed possible to estimate the depth of such a circular edge. A fronto-planar interpolation of that circle will result in a flat circular disc.

Now consider the point at the center of that disc. In 3D space, it will be located at or near

the center of the sphere. If that point is then projected into the other views, the observed normal will always be that of a surface facing the observer. Since all images will thus observe the same gray value, no sensible estimate of the normal will be possible.

9.2 Infeasible Algorithm

Recall the argument behind the denoising depth estimation from chapter 4: instead of taking the gray values seen in the reference image at face value, we treat the true gray value as an additional unknown, and we look for the maximum of the joint likelihood distribution of depth and gray value. This idea can be extended by replacing the gray value in the above argument by a set of shading parameters n and a_d. This means that we would be looking for the most likely combination of n, a_d and the depth z for each pixel. Since the normal is a 2D quantity, while each detector type requires a separate albedo, under one SE and one BSE detector, this results in a 5D parameter set. At each assumed depth, we would therefore need to find the maximum of a 4D-distribution.

One way of doing this would be to use the closed-form normal inference procedure from 8.1. This procedure minimizes an L^2 error, however, so it is not robust to occlusion. It is in fact equivalent to the simplified, non-robust depth estimation that was used for noise calibration and as a didactic example in 4.2.1. There, we formed averages over all observed gray values in order to obtain the most likely true gray value. Only few occluded views that show a vastly different gray value would then suffice to dislocate that estimate. In order to obtain a robust estimate, it was necessary to consider each plausible gray value hypothesis separately. Because of a convenient choice of energy function, it was possible to reduce the set of plausible hypotheses to those within 3 std. deviations of the gray value seen in the reference view, and to then also limit the set of hypotheses that are influenced by each further observation. Ultimately, in my experiment, only 32 gray-value hypotheses were considered at each depth.

If those hypotheses were to represent 4D-combinations of normal and albedo assumptions instead, then the observations in the reference view would no longer allow us to restrict the hypothesis space to a small number of possibilities. Instead, for every hypothetical normal and each detector d, there would be an albedo value a_d that would result in the precise gray value seen in the reference view. For every single normal hypothesis, we would therefore be looking for a 1D value within a certain range of albedos, and we could sample that range using 32 albedo hypotheses. The total number of hypotheses would therefore be equal to 32 times the number of normal hypotheses. There would be no additional information in the reference view with which to further limit the set of plausible normal hypotheses. In order to obtain a normal resolution commensurate with the signal resolution in the image (typically, 256 values), we would require on the order of 100×100 such normal hypotheses.

This corresponds to a 10,000-fold increase in the number of operations. The computation of one denoised depth map from 51 views by the algorithm from chapter 4 took in excess of 14 hours using my naive implementation running on a single CPU of an Intel Xeon E5-1620 running at 3.6 GHz. The limitation to only 51 views was motivated by photoconsistency considerations, since allowing for excessive changes in viewing angle would also lead to large changes in surface radiance for any given point. Since this algorithm would no longer rely on photoconsistency, one would preferrably use all of the available views, i.e. 320 in my experiments. This would lead to an additional increase in the number of operations by a factor of six. Not accounting for the loss in cache performance inherent in managing

the greater number of hypotheses and the increase in computing power over the years, the proposed infeasible algorithm would then require more than 95 years to compute one single depth map.

Although that number may appear daunting at first, the algorithm would still operate on the pixels independently, so it would be perfectly parallelizable. It is therefore likely that further advances in computing technology will make such an algorithm feasible at some point in the near future. Until then, attempts could be made to accelerate the process of fitting the shading parameters robustly to a set of unreliable observations. One possibility would be to use the RANSAC algorithm [91] for that purpose. In my attempts, this has not produced satisfactory results for sufficiently small numbers of random samples, while a larger number also leads to an excessive computation time.

9.3 Feasible Algorithm

The central problem of the above algorithm is the fact that the observed gray values are contaminated by outliers caused by occluding surfaces. If all the observations were reliable, then a linear fit of the unknown shading parameters at each depth would suffice. Fortunately, the two-angle parametrization of rotations obtainable under probably most scanning electron microscopes available today provides us with a situation where this is indeed possible. Recall the precise capture scheme we have been employing starting from chapter 6: the probe can be tilted by up to 60°, while an independent rotation angle allows us to view the tilted probe from different sides. If the tilt angle is set to zero, then, if we neglect the small perspective effects, different rotation angles all produce in-plane rotations of the same 2D view.

While in the context of MVS these rotations are redundant, they do also rotate the probe relative to the detectors. This allows us to estimate the shading parameters from the zero-tilt image sequence alone. Since the sequence shows pure in-plane rotations, all images contain the same geometry, and no occlusions take place. A similar approach was pursued by Pintus et al. [72], though they only recorded four 90° rotations and then used a shading model for symmetrical BSE-detectors. My proposed method uses all available views (20 in my experiments) and both detectors. Furthermore, once the shading parameters are known, my algorithm then predicts the gray value seen at different tilt angles, and it finds the most likely depth using robust, 1D gray-value comparisons.

Formally, the algorithm is defined as follows. One of the zero-tilt images is selected as the reference image i_0. The ray corresponding to each pixel of that image is back-projected into the volume and sampled along discrete intervals, resulting in a set of world-space points P_z. Each P_z is then forward-projected into each zero-tilt view, resulting in a set of gray values $v_{i,d}$, one per image i and detector d. From the $v_{i,d}$, shading parameters a_d and n are estimated using the linearized estimation from 8.1.

Point p_z is then projected into all remaining views j with non-zero tilt, and the observed gray values $v_{j,d}$ are compared to the simulated gray values $\widetilde{v}_{j,d}$ that are given by the model,

$$\widetilde{v}_{j,d} \coloneqq a_d \max(0, p_d \cdot (\bar{V}_j n))^k. \tag{9.1}$$

The difference of the two is then transformed by the Blake-Zisserman energy [76], which corresponds to a superposition of a normal distribution with a uniform one,

$$E_{\mathrm{BZ}}(x) \coloneqq -\log\Big(\exp(-\frac{x^2}{\sigma_{\mathrm{BZ}}^2}) + \tau_{\mathrm{BZ}}\Big). \tag{9.2}$$

The parameters σ^2_{BZ} and τ^2_{BZ} describe the assumed noise variance and tolerance level. The energy contributions of all views j and detectors d then constitute the energy E_z at depth z,

$$E_z \coloneqq \sum_{j,d} E_{\mathrm{BZ}}(\widetilde{v}_{j,d} - v_{j,d}). \tag{9.3}$$

The depth of least energy is then selected as the depth of that pixel, a more precise optimum is determined through quadratic interpolation, and the confidence c of the pixel is computed analogously to 6.1. Finally, the dense depth map is integrated as described in 8.2.

9.4 Experiments

I have tested the algorithm presented above on the image grid used in the experiments in chapters 6 and 8. One of the zero-tilt images was chosen as the reference, the shading parameters at each hypothetical depth were computed from all 20 zero-tilt images, and each such shading hypothesis was evaluated against all images from all 20 rotation angles and all 16 tilt angles. Slices through the resulting energy are shown in Figs. 9.1, 9.2 and 9.3. The resulting sparse depth map was then integrated along the estimated normals. No sky term (mentioned in 5.1) was applied for depth integration, since there are no other views that would cancel out its potentially destructive effects on the depth map. Renderings of the resulting depth map are displayed in Figs. 9.4 and 9.5.

The following parameter values were used:

Normal estimation:	$\tau_\sigma = 0.1$		
Depth estimation:	$\sigma_{\mathrm{BZ}} = 0.04$	$\tau_{\mathrm{BZ}} = 0.01$	
Depth integration:	$\mu_C = 1$	$\lambda_S = 0.02$	$\lambda_N = 0.04$

Figure 9.1: Slices through the robust energy $E_{x,z}$ (left) and the correponding probability $p_{x,z}$ (right) at the indicated y-value (top). Note the very dramatic improvement in depth reliability as compared to the photoconsistency-based cost in Figs. 6.3 and 6.4 on pages 71 and 72. Their quality is indeed similar to that from the narrow-baseline scenario shown in Figs. 4.3 and 4.4 on pages 40 and 41. Additional slices are shown in the following figures.

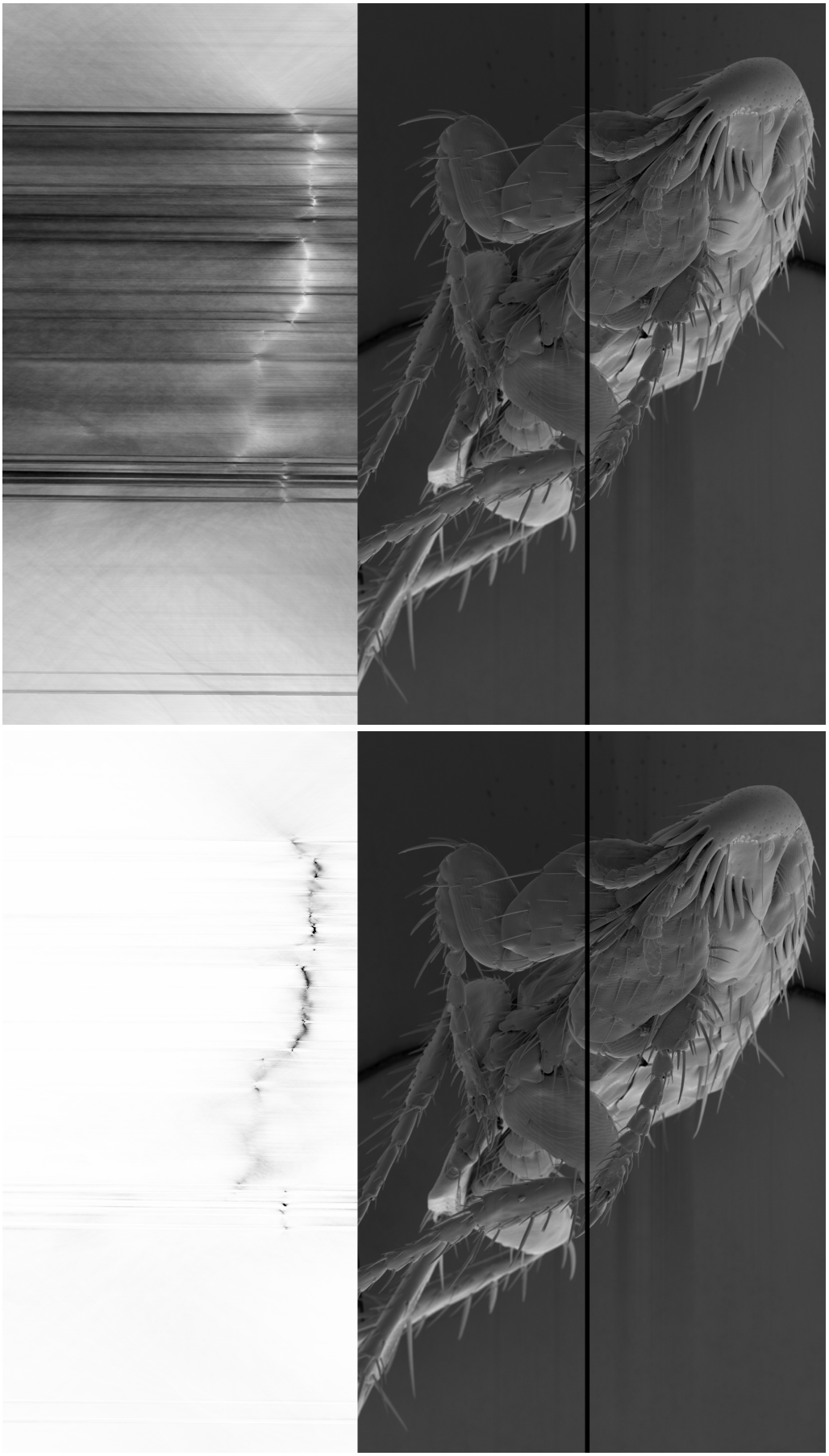


Figure 9.2: Slices analogous to Fig. 9.1.

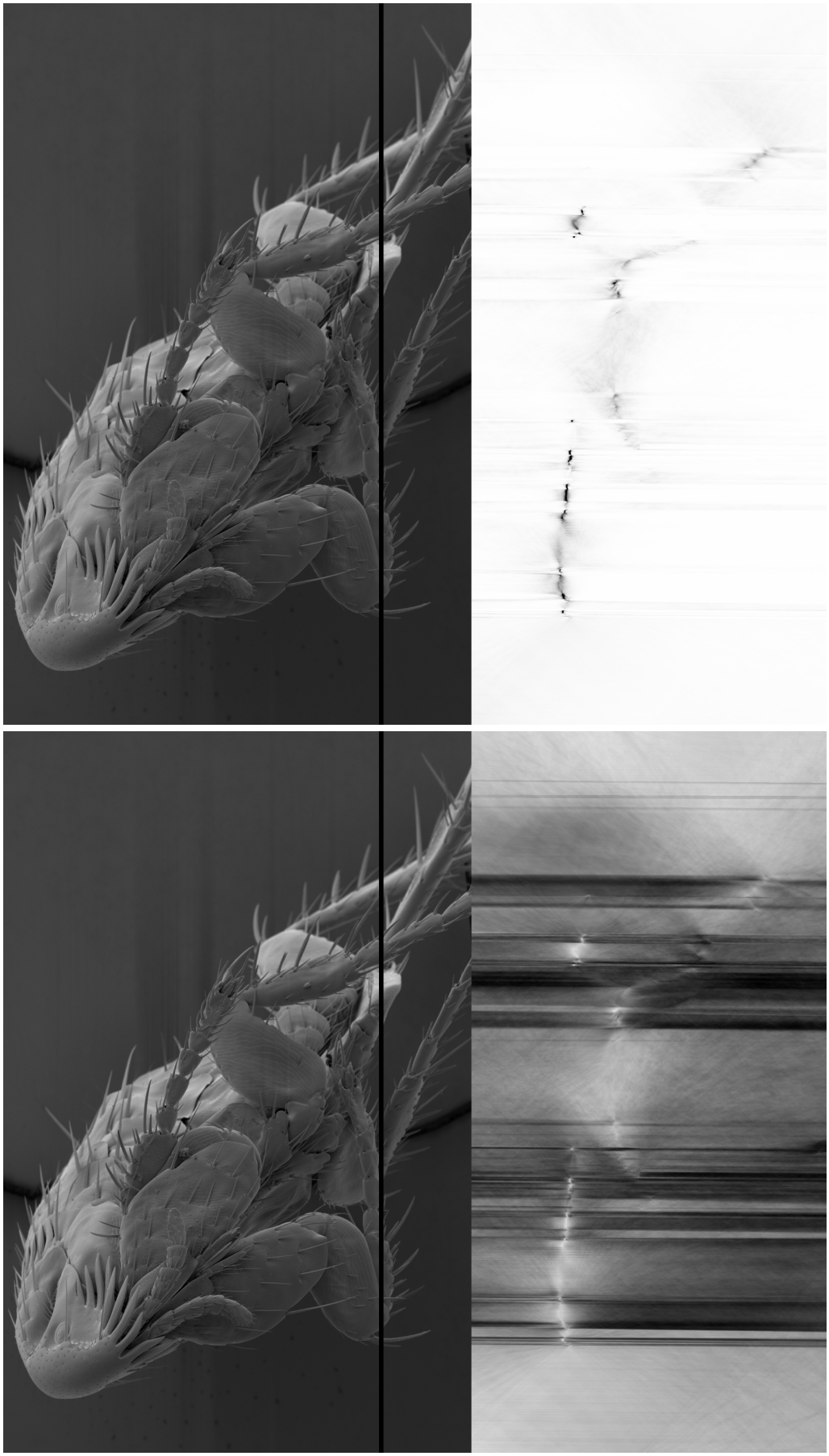


Figure 9.3: Slices analogous to Fig. 9.1.

Figure 9.4: **Left:** SE image taken from the zero-tilt reference view. **Right:** Reconstructed depth map. The background was segmented using the result from chapter 8. Note the greatly improved shape of the head compared to the renderings in Figs. 8.4 and 8.5 on pages 99 and 100.

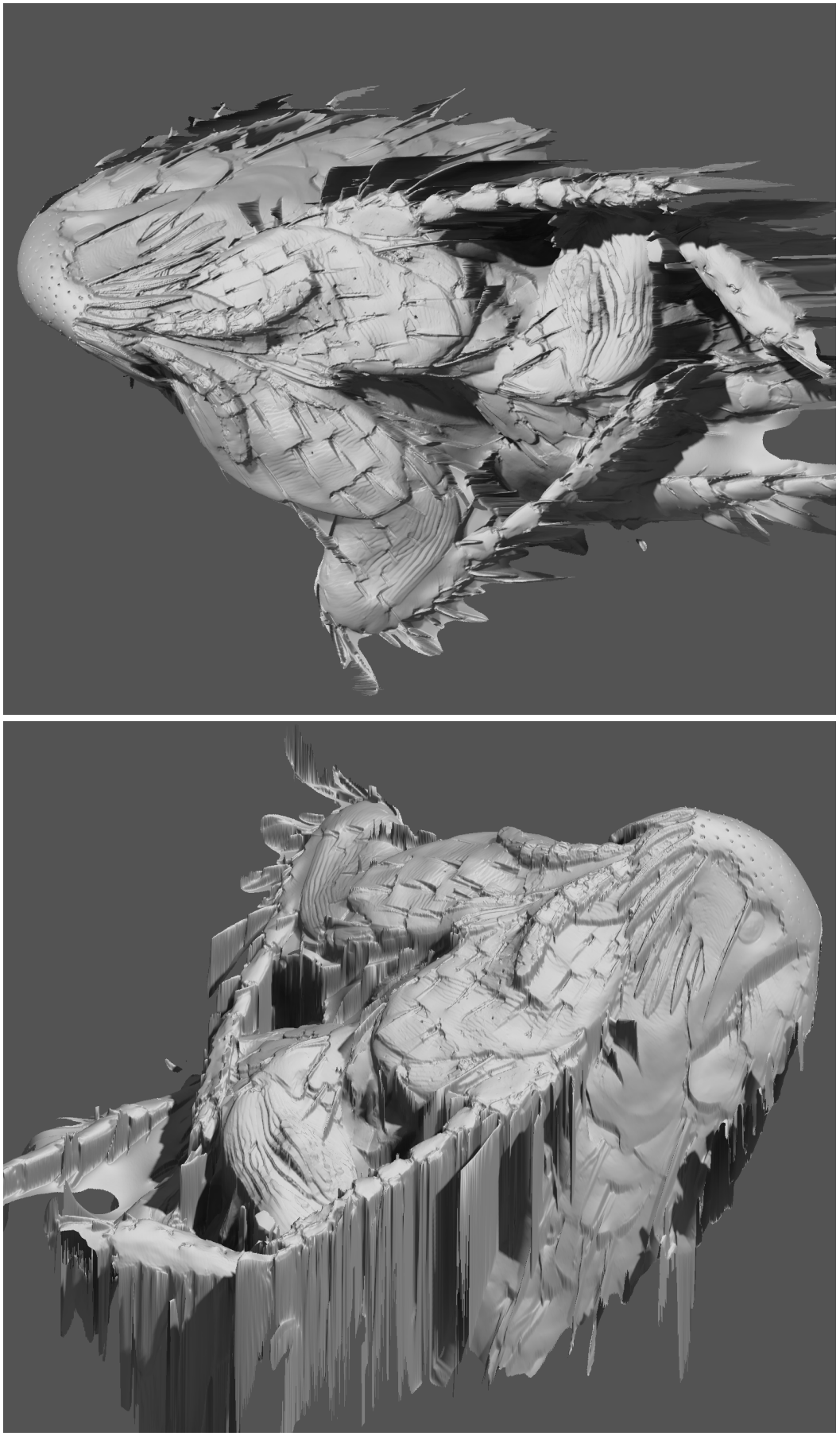


Figure 9.5: The depth map from Fig. 9.4 rendered from other viewpoints. The artifacts around the contours are a consequence of the fact that only a single view is available.

9.5 Conclusions

As can be seen in the slices through the energy volumes and in the renderings themselves, among the presented methods this depth estimation approach is the one that leads to the best results by far. Its drawback is the fact that only a single view can be estimated this way.

Chapter 10

Conclusion and Outlook

I have proposed four different approaches to the reconstruction of intricate 3D surfaces from SEM images. The first approach, discussed in chapters 4 and 5, requires a very dense sampling of viewing angles and is based on photoconsistency. It is able to reconstruct very thin and faint surface elements by simultaneously estimating the depth and a denoised gray value of each pixel. Occlusions are handled through a robust occlusion model, and the resulting non-convex energy function is optimized efficiently by a novel algorithm designed specifically for that energy. The limited range of viewing angles that follows from the dense angular sampling is overcome through a shape model that assumes a locally quadrical surface. This assumption allows it to reconstruct very thin cylindrical features from a small range of viewing angles, though it leads to artifacts in smooth areas. The first approach is the only one that is applicable if a dense image sequence is already given, e.g. if the aim is to colorize an SEM movie sequence.

The second approach, presented in chapter 6, also assumes photoconsistency, but works on a much coarser angular sampling of the scene. This allows for a broader range of viewing angles, but leads to a drastic reduction of the reliability of the estimated depth maps. As a consequence, a surface reconstruction technique had to be developed that is robust to the specific degradations present in those depth maps and that can still capture thin surface elements.

Those first two approaches are not strictly dependent on the SEM imaging modality, which means that they can be applied as-is to optical images. In chapter 7, I have presented a novel empirical shading model for SEM images on which the two remaining approaches are based. This model is specific to SEM, so these approaches are no longer applicable to optical images. Furthermore, since the parameters of the shading model need to be determined for every electron detector that is used, only the first two approaches are applicable if no such calibration can be performed. This could be the case if e.g. the images were captured in the past, and the microscope is no longer accessible, or if its detectors have been configured in a different way in the meantime.

The third approach, presented in chapter 8, combines depth estimates obtained as part of the second approach with surface orientation information obtained through the shading model. The addition of shading information leads to a significant improvement in the reconstruction of smoothly curved areas, but the approach is ultimately limited by the unreliable depth estimates that are part of the second approach.

The fourth approach, proposed in chapter 9, estimates the depths themselves using the shading model. This leads to the most reliable depth maps that my methods could estimate

from a coarse angular sampling. The quality of the depth maps does indeed rival that of the first approach. In contrast to the first approach, the slightly larger number of unknowns involved in shading-based depth estimation precludes the use of a non-convex, robust energy function to deal with occlusions. As a consequence, that depth estimation method is currently only applicable to a single viewing direction that shows the object directly from above, since that is the only direction from which multiple, in-plane rotated images can be obtained. More advanced computing hardware or a more advanced numerical method for the optimization of the non-convex energy would be needed to allow for the estimation of the shading parameters in the presence of occlusions.

Alternatively, a different capture setup would also make the fourth approach applicable for other viewing directions. Two such setups are in existence today, though they were not available for my experiments. First, a stage that allows views to be captured using all six degrees of freedom would make it possible to obtain in-plane rotated views from any given direction. The proposed algorithm could then be applied directly to the resulting images. Second, a more advanced version of the software used to automate the image capture process would facilitate the recording of more than two detector responses at a time. These could e.g. be all three ring segments of the BSE detector and the two SE detectors that were already connected to the microscope I used. These five values would probably allow for the estimation of the shading parameters from a single view, thereby eliminating the need for an in-plane rotated subset of the images. Since this would result in an eight-fold decrease in the number of input variables (1×5 numbers vs. 20×2), this latter setup would require further examination.

As a further alternative, a method could be developed that uses the depth map obtained through the fourth method as an additional input to the regional terms used by the third method. This would then probably improve the quality of the shape resulting from the third method in the areas that are visible in the top view. Special care would have to be taken, however, to deal correctly with overhanging structures in the depth map. Otherwise, this would erroneously fill up cavities such as the ears of the flea that I used in my experiments.

Finally, the geometric calibration of the captured image sequences still forms the main bottleneck for all of the proposed techniques. The primary reason for the very involved manual calibration was the presence of a considerable nonlinear distortion in the images. In my experiments, I used the microscope at or near its lower resolution limit, i.e. to view unusually large objects. It is possible that this distortion is less pronounced when the microscope is used to capture smaller objects. In either case, it is likely that the distortion at any given scale is independent of the object being viewed. If that is the case, then the distortion could be measured once, and the calibration would then consist in finding a rigid-body alignment between the views. It is likely that this could be done in an automated way.

Bibliography

[1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187–194, ACM Press/Addison-Wesley Publishing Co., 1999.

[2] T. J. Cashman and A. W. Fitzgibbon, "What shape are dolphins? building 3d morphable models from 2d images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 232–244, 2013.

[3] S. Romdhani and T. Vetter, "Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 986–993, IEEE, 2005.

[4] B. Amberg, R. Knothe, and T. Vetter, "Expression invariant 3d face recognition with a morphable model," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pp. 1–6, IEEE, 2008.

[5] S. Schönborn, A. Forster, B. Egger, and T. Vetter, "A monte carlo strategy to integrate detection and model-based face analysis," in *German Conference on Pattern Recognition*, pp. 101–110, Springer, 2013.

[6] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, IEEE, 2006.

[7] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, 2010.

[8] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.

[9] O. Faugeras and R. Keriven, *Variational principles, surface evolution, PDE's, level set methods and the stereo problem*. IEEE, 2002.

[10] A. Yezzi and S. Soatto, "Stereoscopic segmentation," *International Journal of Computer Vision*, vol. 53, no. 1, pp. 31–43, 2003.

[11] P. Gargallo, E. Prados, and P. Sturm, "Minimizing the reprojection error in surface reconstruction from images," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[12] S. Liu and D. B. Cooper, "Statistical inverse ray tracing for image-based 3d modeling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 10, pp. 2074–2088, 2014.

[13] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, "Continuous markov random fields for robust stereo estimation," in *European Conference on Computer Vision*, pp. 45–58, Springer, 2012.

[14] P. Labatut, J.-P. Pons, and R. Keriven, "Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[15] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons, "Towards high-resolution large-scale multi-view stereo," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1430–1437, IEEE, 2009.

[16] C. Bailer, M. Finckh, and H. P. Lensch, "Scale robust multi view stereo," in *European Conference on Computer Vision*, pp. 398–411, Springer, 2012.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[19] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. H. Gross, "Scene reconstruction from high spatio-angular resolution light fields.," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[20] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Occluding contours for multi-view stereo," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2014.

[21] K. Yücer, C. Kim, A. Sorkine-Hornung, and O. Sorkine-Hornung, "Depth from gradients in dense light fields for object reconstruction," in *Proceedings of International Conference on 3D Vision (3DV)*, 2016.

[22] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, vol. 21, pp. 163–169, ACM, 1987.

[23] V. Lempitsky and Y. Boykov, "Global optimization for shape fitting," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[24] G. Vogiatzis, P. H. Torr, and R. Cipolla, "Multi-view stereo via volumetric graph-cuts," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 391–398, IEEE, 2005.

[25] G. Vogiatzis, C. H. Esteban, P. H. Torr, and R. Cipolla, "Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2241–2246, 2007.

[26] A. Hornung and L. Kobbelt, "Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 503–510, IEEE, 2006.

[27] S. N. Sinha and M. Pollefeys, "Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 349–356, IEEE, 2005.

[28] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pp. 428–441, IEEE, 2002.

[29] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, (Aire-la-Ville, Switzerland, Switzerland), pp. 61–70, Eurographics Association, 2006.

[30] M. Bolitho, M. Kazhdan, R. Burns, and H. Hoppe, "Parallel poisson surface reconstruction," in *International symposium on visual computing*, pp. 678–689, Springer, 2009.

[31] K. Zhou, M. Gong, X. Huang, and B. Guo, "Data-parallel octrees for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 5, pp. 669–681, 2011.

[32] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, p. 29, 2013.

[33] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[34] T. F. Chan and S. Esedoglu, "Aspects of total variation regularized l 1 function approximation," *SIAM Journal on Applied Mathematics*, vol. 65, no. 5, pp. 1817–1837, 2005.

[35] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l 1 range image integration," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[36] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *Journal of Mathematical Imaging and vision*, vol. 28, no. 2, pp. 151–167, 2007.

[37] K. Kolev, M. Klodt, T. Brox, S. Esedoglu, and D. Cremers, "Continuous global optimization in multiview 3d reconstruction," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 441–452, Springer, 2007.

[38] K. Kolev, M. Klodt, T. Br ox, and D. Cremers, "Continuous global optimization in multiview 3d reconstruction," *International Journal of Computer Vision*, vol. 84, no. 1, pp. 80–96, 2009.

[39] D. Cremers and K. Kolev, "Multiview stereo and silhouette consistency via convex functionals over convex domains," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1161–1174, 2011.

[40] K. Kolev, T. Brox, and D. Cremers, "Fast joint estimation of silhouettes and dense 3d geometry from multiple images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 493–505, 2012.

[41] K. Kolev, T. Pock, and D. Cremers, "Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo," in *European Conference on Computer Vision*, pp. 538–551, Springer, 2010.

[42] J. Weickert, *Anisotropic diffusion in image processing*, vol. 1. Teubner Stuttgart, 1998.

[43] J. Weickert, "Coherence-enhancing diffusion filtering," *International journal of computer vision*, vol. 31, no. 2-3, pp. 111–127, 1999.

[44] A. M. Mendrik, E.-J. Vonken, A. Rutten, M. A. Viergever, and B. van Ginneken, "Noise reduction in computed tomography scans using 3-d anisotropic hybrid diffusion with continuous switch," *IEEE Transactions on Medical Imaging*, vol. 28, no. 10, pp. 1585–1594, 2009.

[45] C. Schroers, H. Zimmer, L. Valgaerts, A. Bruhn, O. Demetz, and J. Weickert, "Anisotropic range image integration," in *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pp. 73–82, Springer, 2012.

[46] T. Rindfleisch, "Photometric method for lunar topography.," 1966.

[47] B. K. Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," 1970.

[48] B. K. Horn, "Understanding image intensities," *Artificial intelligence*, vol. 8, no. 2, pp. 201–231, 1977.

[49] F. Nicodemus, J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis, "Geometrical considerations and nomenclature for reflectance. us department of commerce 1977," *Online erhältlich unter http://graphics. stanford. edu/courses/cs448-05-winter/"-papers/nicodemus-brdf-nist. pdf Eingesehen am*, vol. 8, 2015.

[50] Klett, E. Witwe, Detleffsen, C. Peter, *et al.*, *IH Lambert... Photometria sive de mensura et gradibus luminis, colorum et umbrae*. sumptibus viduae Eberhardi Klett, 1760.

[51] M. Oren and S. K. Nayar, "Generalization of the lambertian model and implications for machine vision," *International Journal of Computer Vision*, vol. 14, no. 3, pp. 227–251, 1995.

[52] R. J. Woodham, "Photometric stereo: A reflectance map technique for determining surface orientation from image intensity," in *22nd Annual Technical Symposium*, pp. 136–143, International Society for Optics and Photonics, 1979.

[53] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[54] J.-D. Durou, M. Falcone, and M. Sagona, "Numerical methods for shape-from-shading: A new survey with benchmarks," *Computer Vision and Image Understanding*, vol. 109, no. 1, pp. 22–43, 2008.

[55] S. Herbort and C. Wöhler, "An introduction to image-based 3d surface reconstruction and a survey of photometric stereo methods," *3D Research*, vol. 2, no. 3, pp. 1–17, 2011.

[56] K. Ikeuchi, "Constructing a depth map from images.," tech. rep., DTIC Document, 1983.

[57] K. Ikeuchi, "Determining a depth map using a dual photometric stereo," *The International journal of robotics research*, vol. 6, no. 1, pp. 15–31, 1987.

[58] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi, "Efficiently combining positions and normals for precise 3d geometry," in *ACM transactions on graphics (TOG)*, vol. 24, pp. 536–543, ACM, 2005.

[59] J. Zivanov, P. Paysan, and T. Vetter, "Facial normal map capture using four lights–an effective and inexpensive method of capturing the fine scale detail of human faces using four point lights," in *In Proc. GRAPP*, Citeseer, 2009.

[60] G. Vogiatzis, C. Hernandez, and R. Cipolla, "Reconstruction in the round using photometric normals and silhouettes.," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 1847–1854, IEEE, 2006.

[61] C. H. Esteban, G. Vogiatzis, and R. Cipolla, "Multiview photometric stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 548–554, 2008.

[62] G. Piazzesi, "Photogrammetry with the scanning electron microscope," *Journal of Physics E: Scientific Instruments*, vol. 6, no. 4, p. 392, 1973.

[63] A. Zolotukhin, I. Safonov, and K. Kryzhanovskii, "3d reconstruction for a scanning electron microscope," *Pattern recognition and image analysis*, vol. 23, no. 1, pp. 168–174, 2013.

[64] M. Eulitz and G. Reiss, "3d reconstruction of sem images by use of optical photogrammetry software," *Journal of structural biology*, vol. 191, no. 2, pp. 190–196, 2015.

[65] A. P. Tafti, J. D. Holz, A. Baghaie, H. A. Owen, M. M. He, and Z. Yu, "3dsem++: Adaptive and intelligent 3d sem surface reconstruction," *Micron*, vol. 87, pp. 33–45, 2016.

[66] K. Ikeuchi and B. K. Horn, "Numerical shape from shading and occluding boundaries," *Artificial intelligence*, vol. 17, no. 1-3, pp. 141–184, 1981.

[67] J. Lebiedzik, "An automatic topographical surface reconstruction in the sem," *Scanning*, vol. 2, no. 4, pp. 230–237, 1979.

[68] L. Reimer and D. Stelter, "Monte-carlo calculations of electron-emission at surface edges," *Scanning Microscopy*, vol. 1, no. 3, pp. 951–962, 1987.

[69] W. Beil and I. Carlsen, "Surface reconstruction from stereoscopy and "shape from shading" in sem images," *Machine vision and applications*, vol. 4, no. 4, pp. 271–285, 1991.

[70] T. Vynnyk, T. Schultheis, T. Fahlbusch, and E. Reithmeier, "3d-measurement with the stereo scanning electron microscope on sub-micrometer structures," *Journal of the European Optical Society-Rapid publications*, vol. 5, 2010.

[71] J. Paluszyński and W. Slowko, "Surface reconstruction with the photometric method in sem," *Vacuum*, vol. 78, no. 2, pp. 533–537, 2005.

[72] R. Pintus, S. Podda, and M. Vanzi, "An automatic alignment procedure for a four-source photometric stereo technique applied to scanning electron microscopy," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 989–996, 2008.

[73] V. Estellers, J.-P. Thiran, and M. Gabrani, "Surface reconstruction from microscopic images in optical lithography," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3560–3573, 2014.

[74] R. Danzl and S. Scherer, *Integrating shape from shading and shape from stereo for variable reflectance surface reconstruction from SEM images*. Citeseer, 2001.

[75] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[76] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[77] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian conference on Image analysis*, pp. 363–370, Springer, 2003.

[78] C. Schroers, S. Setzer, and J. Weickert, "A variational taxonomy for surface reconstruction from oriented points," in *Computer Graphics Forum*, vol. 33, pp. 195–204, Wiley Online Library, 2014.

[79] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[80] M. R. Oswald and D. Cremers, "Surface normal integration for convex space-time multi-view reconstruction.," in *BMVC*, 2014.

[81] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011.

[82] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Proc. ISPRS intercommission conference on fast processing of photogrammetric data*, 1987.

[83] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, ACM, 1996.

[84] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *IEEE Proc. Conf. On*, 1997.

[85] D. Claus and A. W. Fitzgibbon, "A rational function lens distortion model for general cameras," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 213–219, IEEE, 2005.

[86] L. Reimer, "Scanning electron microscopy: physics of image formation and microanalysis," 2000.

[87] B. T. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[88] J. F. Blinn, "Models of light reflection for computer synthesized pictures," in *ACM SIGGRAPH Computer Graphics*, vol. 11, pp. 192–198, ACM, 1977.

[89] E. P. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg, "Non-linear approximation of reflectance functions," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 117–126, ACM Press/Addison-Wesley Publishing Co., 1997.

[90] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[91] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.