TECHNISCHE
UNIVERSITÄT
DARMSTADT

GCC
Graphics, Capture and
Massively Parallel Computing

Master's Thesis

Reconstruction of Specular Surfaces from Reflectance Correspondences

Stepan Konrad

February 2016

Technische Universität Darmstadt
Department of Computer Science
Graphics, Capture and Massively Parallel Computing

Supervisor: Prof. Dr.-Ing. Michael Goesele

Thesis Statement pursuant to § 22 paragraph 7 of APB TU Darmstadt

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper. I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content. This thesis has not been handed in or published before in the same or similar form. In the submitted thesis the written copies and the electronic version are identical in content.

Darmstadt, 15.02.2016

Abstract

Image-based reconstruction of specular surfaces usually requires dense correspondences between image features and points in the environment. In natural environments, these points are usually unknown and correspondences often exist only sparsely between pairs of images. These assumptions complicate the reconstruction problem by introducing many ambiguities which can often only be resolved using regularization of the surface. Only very recently, work has been presented which is able to reconstruct specular surfaces using different kinds of algorithms.

This thesis gives an introduction to the different types of ambiguities and presents a framework which tries to resolve these through regularization using a multi-view approach in combination with a low-parametric surface. The reconstruction method is modeled as an iterative optimization in order to achieve specular consistency. This consistency is based on the laws of reflection applied to the viewing rays which are given by image-to-image features. The framework is capable of processing different kinds of additional input data, e.g. known environmental features or boundary points on the surface.

Synthetic and real-world experiments were executed using both known and unknown feature positions. Results on synthetic datasets show accurate reconstructions even in the presence of specular consistent ambiguities. An adapted outlier removal for feature matching on image series of specular objects was applied to real-wold input data. The results show that it is possible to reconstruct the surface of mirroring objects even with sparse input data.

Acknowledgements

I would like to express my very great appreciation to Michael Goesele, Richard Szeliski and Sudpita Sinha for their excellent support and insightful comments during the research time of this thesis. The research meetings of the CR-PLAY project (`www.cr-play.eu`) were a great place to receive instructive feedback and I would like to thank all members of these meetings for their great support. In addition, I would like to thank every member of the Graphics, Capture and massively parallel Computing group individually who encouraged and helped me during the work on this thesis.

Contents

1 Introduction

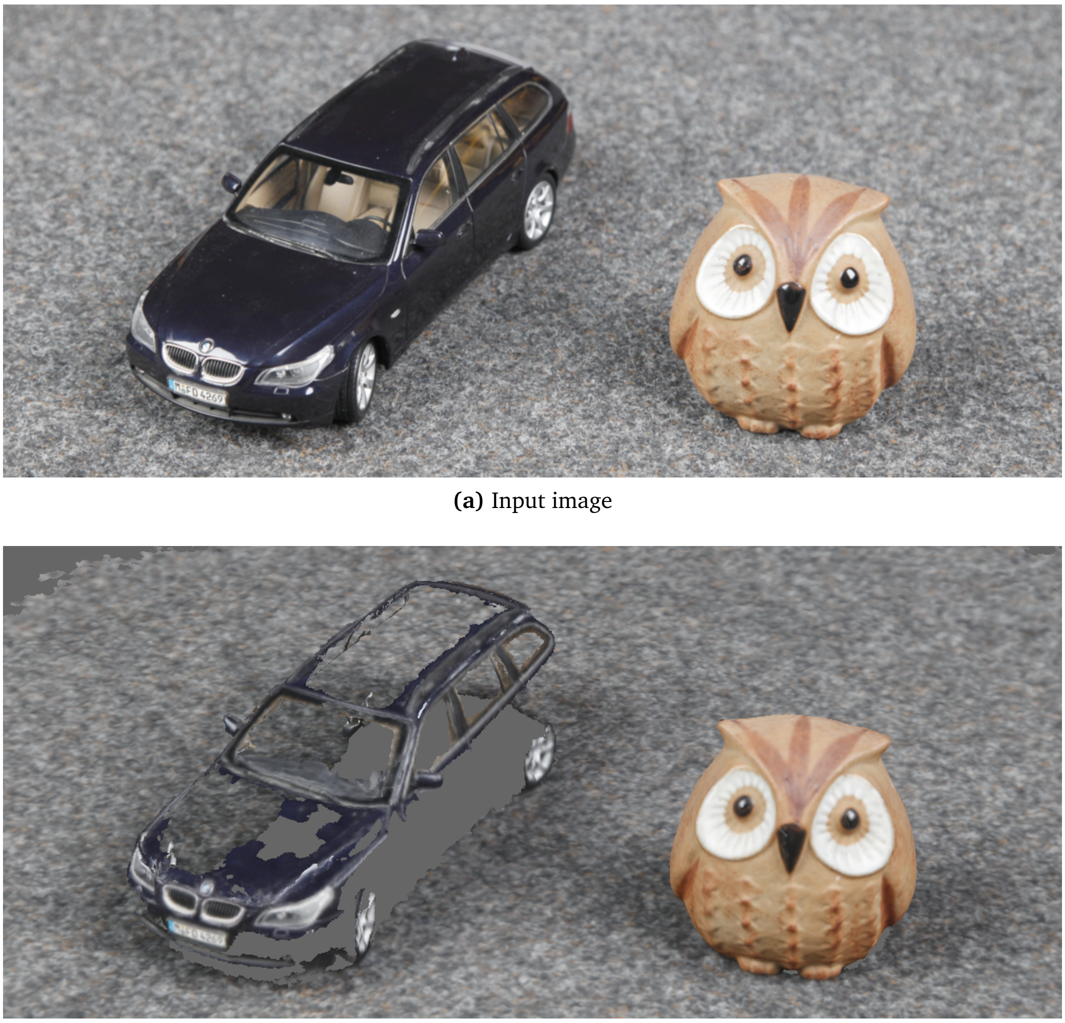
The rise of digital imaging motivated people to capture and preserve their environment. Scientific achievements in the research field of 3D reconstruction delivered powerful tools to create accurate models of objects using photographs. Over the last decade these tools have been further developed to be able to reconstruct scenes with a city or world scale [Agarwal et al., 2009]. Moreover, it is possible to reconstruct geometry of famous sights with high accuracy even from tourist photographs [Snavely et al., 2006; Goesele et al., 2007].

However, standard algorithms based on structure from motion (SfM) and multi-view stereo (MVS) techniques often fail on some categories of objects common in natural and man-made environments. These categories include objects consisting of refractive or reflective materials, e.g. water, windows, mirrors, glossy art objects, metal-coated objects or cars. The cause for this failure is that these algorithms use the simplified assumption that the scene is only made of diffuse and opaque materials.

The intrinsic principle of purely transparent and mirroring objects is that the perceived color information at a particular point on the surface is only either a transmitted or reflected color information from the surrounding. It is not possible to see such an object *directly*; humans infer the geometry by interpreting how it distorts the environment.

This principle makes it impossible for traditional SfM-based algorithms to reconstruct this class of objects. The reason for this is the inability to establish correspondences between projections of a certain point on the surface in two images. Thus, in natural scenes with unknown illumination the position of this point cannot be estimated through triangulation. Triangulating points from two or more measurements relies on epipolar geometry. If the viewing rays which are used for triangulation are reflected on the surface, this geometric property does not hold any more. This implies that the specular objects are either reconstructed at a wrong location, have distorted geometry or are missing entirely in the reconstruction of a scene (see Figure 1.1).

In the last decade there has been a lot of research with the aim to reconstruct specular, glossy or transparent objects. The reconstruction problem has been modeled using



(a) Input image

(b) Rendering of 3D reconstruction

Figure 1.1: *Top:* One of about 60 input images taken of a specular model car and a ceramic owl decoration which is mostly diffuse. *Bottom:* Rendering of the result from a standard image-based reconstruction pipeline [Fuhrmann et al., 2015]. The specular object has many missing parts on the surface.

various approaches which assume different kinds of input data (see Section 2.3). The majority of these approaches lead to very accurate results. However, they use active illumination or very constraint lab settings in order to establish dense correspondences between the light sources or pattern generators and the visible response that is observed by the camera. Even if only sparse correspondences are required, the positions of the light source or the pattern have to be known.

Only very few publications were recently shown that allow for a natural, unknown illumination. Still, these approaches either require the capture of specific objects [Li et al., 2014], specific capture setups [Jacquet et al., 2013] or the acquisition of an environment map [Godard et al., 2015].

The contribution of this thesis is the reconstruction of specular surface geometry using sparse reflectance correspondences from images or video. The objects are assumed to be captured in static, natural scenes. Usually this means that there is enough diffuse geometry in the environment to estimate the camera positions correctly. Furthermore, the objects should consist of a material with high specularity (approximating a perfect mirror) to be able to match reflected image features between the input images. It should be noted that the 3D locations of these image features do not have to be known but can be estimated jointly during the reconstruction.

The remainder of this thesis is organized as follows: Chapter 2 explains the foundations of specular geometry and presents significant work in the research field of the reconstruction of specular objects. In Chapter 3, a novel approach for reconstructing specular objects from images in unknown environments is presented. Chapter 4 presents experiments with synthetic and real-world data. In Chapter 5, the insights from the previous chapter are evaluated and a prospect for future work is presented.

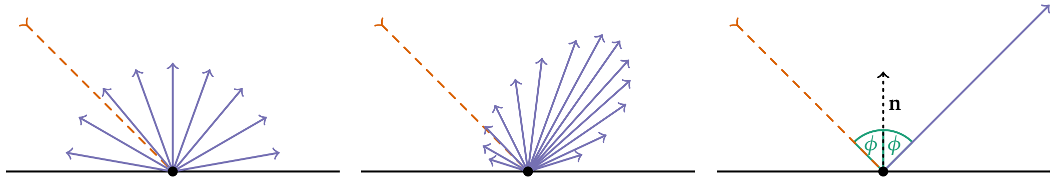
2 Foundations and Related Work

2.1 Specular Objects

Diffuse surfaces have in common that they reflect incoming light into a broad range of directions. In contrast, a pure or highly specular reflection is defined by reflecting incoming light rays into a single outgoing direction. Furthermore, the angle of incident light is the same as the angle of reflection with respect to the surface normal. We can see this relation in Figure 2.1. A glossy (or specular) material reflects light mainly in the direction of pure specular reflection, however, light is scattered in other directions, as well.

2.1.1 Physical Properties

Reflectance describes the ratio of radiance that is reflected by a surface divided by the incoming radiance which is received by the surface. This reflectance is not constant for a specific material but often varies at different wavelengths. For example, the reflectance of gold is lower in the blue wavelengths resulting in a more yellowish appearance. In addition, real-world objects often consist of materials which have a mixture of specular and diffuse reflective properties. On diffuse surfaces, the reflectance is uniform leading



(a) Diffuse reflection **(b)** Glossy/specular reflection **(c)** Pure specular reflection

Figure 2.1: Different materials reflect light in different ways. This illustration shows example light distributions for diffuse and specular/glossy materials. For pure/perfect specular reflection, the incoming light direction (orange dashed) and the single reflection direction (blue) intersect with the surface at the same angle ϕ with respect to the surface normal **n** (dotted).

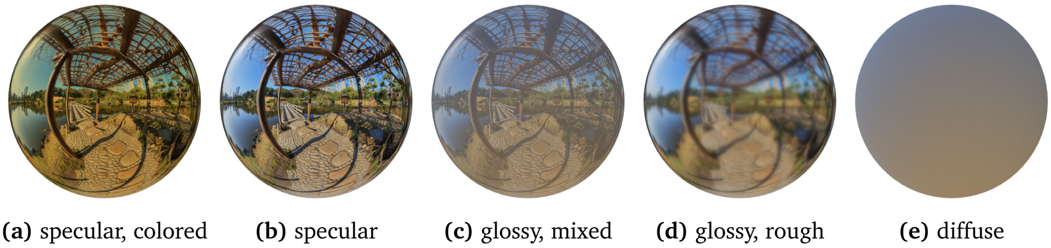


Figure 2.2: Renderings of spheres with different materials. The materials of the first three spheres result in sharp reflections of the environment. Lit using Smart IBL from HDRLabs.com/sibl.

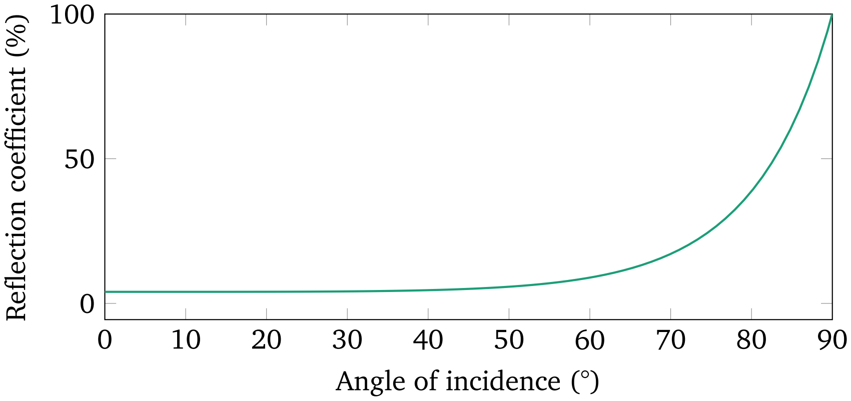


Figure 2.3: The reflection coefficient with respect to the angle of incidence from air to glass. The plot shows the reflection coefficient only for unpolarized light.

to an almost identically reflected radiation in all directions. The reflectance of highly specular surfaces, however, is non-zero only in a single direction. This makes it possible to see sharp reflections of an environment through a specular surface. Figure 2.2 shows a rendering of spheres with different materials in the same environment. We can only capture sharp and detailed reflections with materials like the ones shown in **(a)** to **(c)**. Those materials produce a high distinctness of image (DOI).

2.1.2 Photographing Specular Reflections

The acquisition procedure for specular objects depends on the surface materials as well as the nature of the environment. We have to set the camera settings in such a way that the specular reflections and the object are well exposed. The Fresnel equations show the correlation between the reflection coefficient and the angle of incident light (see Figure 2.3). We have to adjust the exposure to accommodate for the loss of reflectance at angles less than 90° for most materials which are no perfect mirror. It may be required

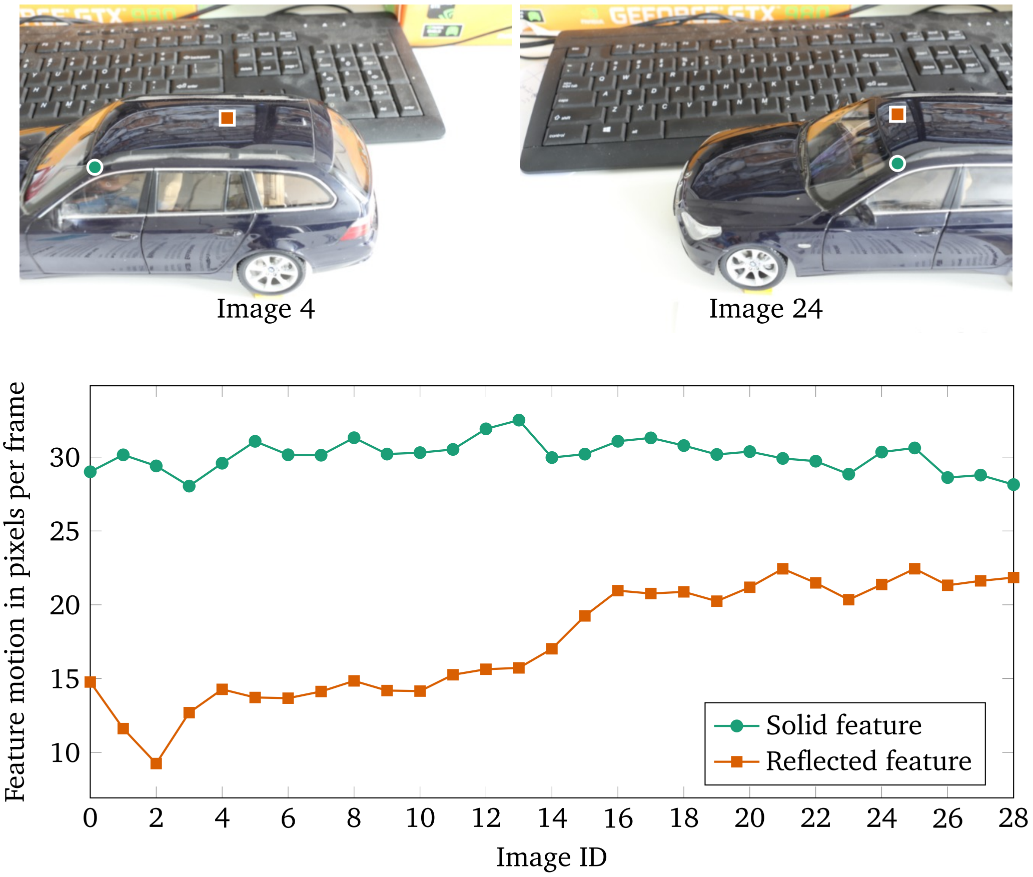


Figure 2.4: The plot shows the amount of motion in pixels on the image plane from one input frame to another for two image features. The green (round) feature lies on the surface of the model car while the orange (square) feature is a reflection of the environment. The camera is moving almost linearly from right to left without rotating. The feature motion is corrected by the amount of camera motion. The amount of motion of the solid feature stays almost constant in contrast to the reflected feature. A higher curvature (the front of the car roof) leads to more similar motion of the different feature types (image 16 to image 28).

to capture multiple images as different exposures if the dynamic range of the scene including the reflections is higher than the maximum dynamic range of the camera. We can then combine the images to a high dynamic range (HDR) image for the subsequent steps [Mann and Picard, 1994; Robertson et al., 1999].

Based on the curvature of the surfaces, the feature reflections can move over the surface rapidly compared to the camera motion. Thus, we need to capture many images while moving or rotating the camera. In Figure 2.4, we can observe a specular feature which moves at different speeds over the image plane because the reflection takes place at surface parts with different curvature. In this example, the camera is only translated

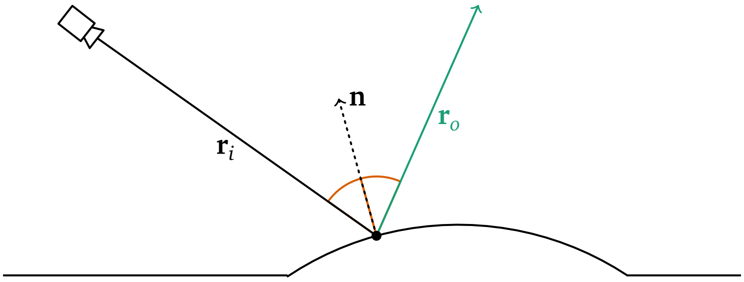


Figure 2.5: The forward problem of specular reflections: The reflection direction (green) can be calculated if the surface normal (dotted) is known because the angle of the incoming ray direction is the same as the angle of the reflection with respect to the normal.

without any rotation. If the surface part has a very high positive curvature (in the direction of the camera movement), the motion vector of reflected features is almost the same as the motion vector of solid features on the surface. The difference of the motion vectors increases with lower or even negative curvatures. However, in practice, motion analysis is more complicated because the camera often also undergoes a rotation.

2.2 Principles of Reflections

Up to now, only physical properties of specular surfaces were outlined. The main ideas of the reconstruction algorithm presented in Chapter 3 are based on the geometric principles of reflections. The following sections introduce these principles and show the state of the art regarding shape from reflections.

2.2.1 Forward Problem

Given the geometry of a specular surface, we can quite easy map image pixels to the direction in the environment from where the color information of this pixel is received. This can be seen as a *forward problem* and can be solved by tracing a viewing ray from the camera center through this pixel into the scene. If this ray hits a mirroring surface, we can use the law of reflection to calculate the outgoing ray direction $\mathbf{r}_o$ based on the normal $\mathbf{n}$ of the surface at the point of intersection and the incoming ray direction $\mathbf{r}_i$ (see also Figure 2.5):

$$\mathbf{r}_o = 2\langle \mathbf{n}, \mathbf{r}_i \rangle \mathbf{n} - \mathbf{r}_i \tag{2.1}$$

2.2.2 Inverse Problem

Considering that we know the incoming viewing ray $\mathbf{r}_i$, the outgoing reflection ray $\mathbf{r}_o$ as well as the point where the rays hit the surface (in other words, we do not know only the directions of the rays but also their origin), we can calculate the normal as easy as in the *forward problem* again by using the law of reflection: All three vectors lie in the same plane and thus the normal vector $\mathbf{n}$ is the bisector of the two rays:

$$\mathbf{n} = \frac{1}{2}(-\mathbf{r}_i + \mathbf{r}_o) \tag{2.2}$$

Thus, this can be seen as the trivial case of the *inverse problem* of finding the surface based on the viewing and reflection rays. However, if we know only the direction of the reflection ray or the position of the point in the environment, we cannot calculate the position S where the viewing ray hits the surface. Because of the unknown depth of the surface, there is a 1-dimensional solution space [proven in Balzer, 2008] of possible normals for each point S along the viewing ray.

Balzer and Werling [2010] discuss this problem in more depth in a survey of different specular surface reconstruction techniques. In this survey the problem is considered mathematically ill-posed because for each viewing ray three unknowns (the distance of the surface from the camera center and the slant and tilt of the normal) have to be estimated as opposed to two scalars that are induced by the law of reflection.

The authors also show that the use of multiple measurements using two (or more) cameras can result in two normal fields that are solely consistent at one specific surface distance (see Figure 2.6). This normal consistency resolves the ambiguity of the 1-dimensional solution space. Tarini et al. [2005] show that consistent depth an normals can be achieved using a stable optimization by minimizing the surface incoherence which arises from normal map integration along different paths. However, the normal consistency or the surface coherence can only resolve the ambiguity for capture setups which are designed to produce dense feature correspondences. If the number of features is rather low resulting in sparse correspondences, the problem has to be regularized. This can be achieved through a parametric surface model with a low number of surface parameters. An alternative approach would incorporate a regularization term into an optimization on regular surface meshes.

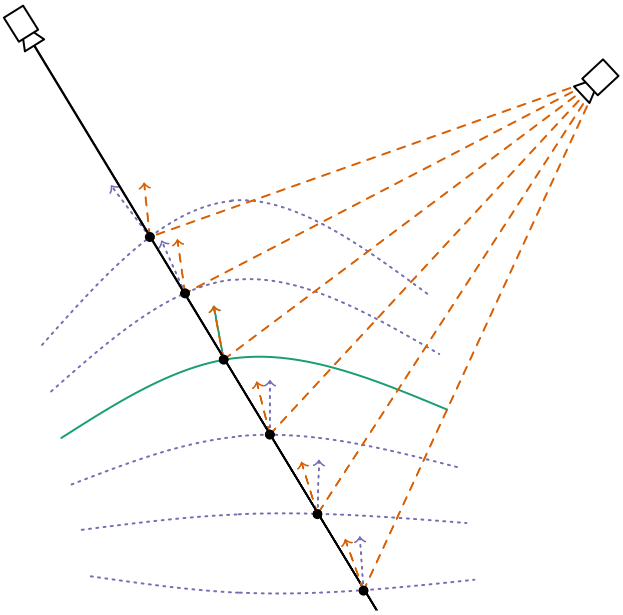


Figure 2.6: For each possible depth along a viewing ray, we can calculate a surface point with a correct normal (blue dotted) using known feature directions. According to Balzer and Werling [2010], a second camera (orange dashed viewing rays and normals) can be used to resolve the 1-dimensional ambiguity by extracting the correct depth and surface normal (green).

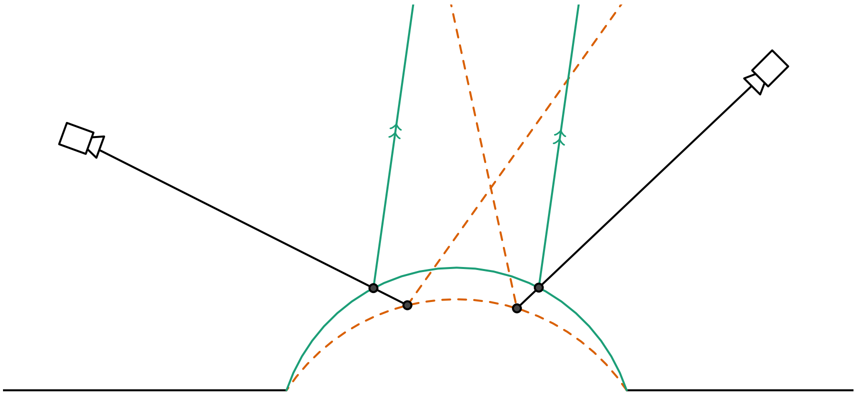


Figure 2.7: This illustration shows the basic principle of surface reconstruction using specular consistency. In contrast to the wrong surface (orange dashed) the correct surface (green) leads to parallel (specularly consistent) reflection rays.

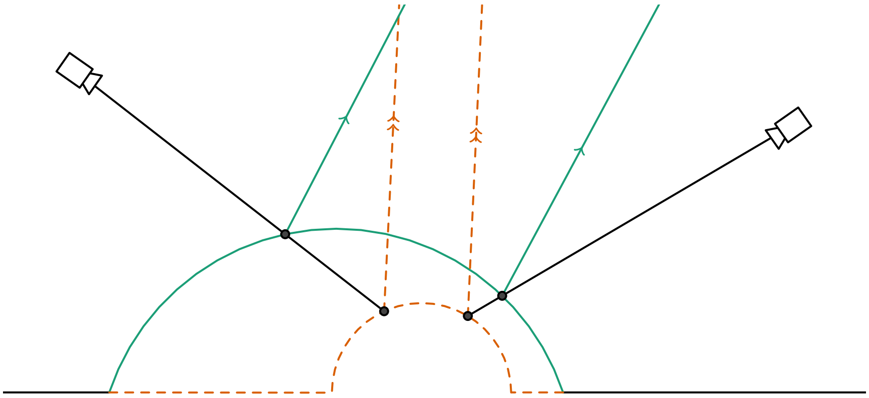


Figure 2.8: This example depicts two viewing rays which reflect to a common environmental feature. There are infinitely many surfaces (e.g. the orange dashed surface) which lead to a specular consistency but have a different shape than the true surface (green).

2.2.3 Inverse Problem with Unknown Environment

The previous problem becomes very challenging if the direction of the reflection ray is completely unknown. This is typically the case if the environment is uncontrolled and thus correspondences between image pixels and environment points cannot be established. The hypothesis which is established in this thesis is that we can use the consistency, which was needed in the solution to the *inverse problem* (hereinafter named *specular consistency*), again here to resolve the ambiguities which were based on the unknown surface depth. This basic principle is depicted in Figure 2.7.

The unknown environmental feature directions naturally lead to many ambiguities: There are infinitely many configurations of surface shape and environmental feature position that obey the law of reflection (see Figure 2.8 for an example). The increased number of unknowns, however, raises the need for a stronger regularization of the reconstructed surface. From the example in Figure 2.9 we can see that a higher number of surface parameters can lead to a local adaption of the measurements in contrast to a globally correct solution. In this example, the problem arises because both viewing rays influence different control points instead of a common point.

2.3 Techniques for Specular Surface Reconstruction

Early foundations date back to Blake [1985] as well as Blake and Brelstaff [1988] who describe how information about local surface geometry can be inferred using specularities. They show that in a traditional stereo setting the triangulation of specularities

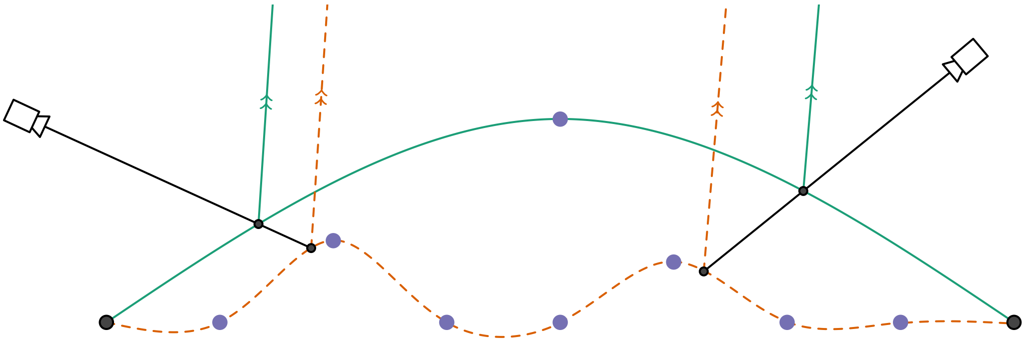


Figure 2.9: In this configuration two cameras see a common feature. There are two B-spline surfaces with a different number of control points (blue) shown as an example. Both surfaces are specularly consistent, i.e. the reflected rays are parallel. The high-parametric surface adapts more locally to the reflection rays.

appears to lie in front of concave surfaces but behind convex ones. They are able to infer curvature and depth information given a known surface point in combination with traditional stereo and monocular specularity analysis (i.e. the shape of the observed specularity). Zisserman et al. [1989] further investigate how specular reflections tracked through multiple views can be used to classify the surface as convex or concave without knowing the light direction. Furthermore, they are able to estimate the depth of the specular surface from the new viewpoint if the depth for the old viewpoint is known.

Subsequently a lot of research has been done especially in the last few years that tries to solve the inverse problem in various ways. Many of these publications assume dense feature correspondences and thus the experiments were performed using structured lighting (see Section 2.3.1). Only very recently, work has been presented (see Sections 2.3.2 and 2.3.3) that is able to reconstruct specular surface geometry in lab settings with unknown environment or even in real-wold scenes with unknown illumination.

2.3.1 Structured Light

The approach presented by Nehab et al. [2008] uses a dense stereo framework based on image intensities to reconstruct glossy objects. These objects (for example a coin) are not highly specular and are illuminated by a nearby LCD. They also use the term “specular consistency” and show how to overcome ambiguities arising from false matches that are nevertheless consistent.

Very accurate reconstructions of large specular surfaces, such as a car engine hood, were shown by Balzer et al. [2011]. They solve the inverse problem by introducing the first multi-view specular stereo algorithm that is capable of resolving the previously pointed out surface ambiguities. Dense correspondences are achieved by using an LCD as a pattern generator.

Recent research from Liu et al. [2015] shows that it is possible to reconstruct a specular surface even from a single image using dense correspondences. They model the reconstruction as a solution to a system of partial differential equations (PDEs) and present a technique to estimate the depth for one point which resolves the surface ambiguities. Moreover, they extend the approach to be able to cope with the relaxed problem using only sparse correspondences. Therefore, they model the surface using uniform cubic B-splines (instead of the solution of PDEs) to regularize the depth estimation. The results show that their formulation of the reconstruction as an optimization problem is capable of accurately estimating the surface geometry even from sparse correspondences.

An older work from Solem et al. [2004] falls in a category between controlled lighting and unknown environment. They need to know the position of a bright light source, e.g, the sun for outdoor scenes, and estimate this position by making use of the shadow which is cast by the camera. Similar to this thesis, their approach tries to find a smooth surface through a set of given points that also minimizes an error based on the surface normal. In a PDE-based approach they model the surface as a time dependent implicit representation and re-calculate the intersection point of the viewing ray with the surface during the optimization process. Using this approach, they were able to reconstruct a car window and a wine bottle successfully.

2.3.2 Unknown Environment

The previous approaches have in common that they require the environment or more precisely the mapping between pixels and 3D locations in the environment to be known. In contrast, Roth and Black [2006] show that it is possible to extract optical flow from image sequences of specular objects using probabilistic inference. Based on the specular flow they are able to reconstruct the actual surface geometry.

Sankaranarayanan et al. [2010] also do not require a known environment. They use SIFT to establish feature correspondences and use a quadric model for the surface geometry. By solving a linear system for surface gradients followed by a Poisson surface integration they can reconstruct highly specular surfaces. However, their work is limited to lab settings as the camera movement is constrained to rotation.

2.3.3 Real-World Settings

Only few publications in the research field of specular surface reconstruction have taken the reconstruction outside of the lab. Jacquet et al. [2013] propose an approach to reconstruct normal maps from nearly flat reflective surfaces. The required input data are videos of mainly glass façades of large buildings. In their approach they use point-to-line instead of point-to-point correspondences to constrain the surface normals. This kind of line reflection tracking allows for dense scanning of the reflective surface. They estimate the camera motion from the rectified input frames using a line tracking approach, as well. After the tracking, they estimate the 3D world positions of the lines. The now known line positions enable to set up constraints on the normals of the surface. For the evaluation of the results and due to the lack of a ground truth, they render the reconstructed scene and evaluate the appearance of the reflections visually. They present visually persuasive results but the reconstructions assumes that on the one hand the surfaces have to be near-planar and on the other hand reflected lines are necessary in the input images.

An approach to reconstruct objects with a more complex geometry has been shown by Li et al. [2014]. In addition, the presented algorithm is able to reconstruct the surface from a single image. However, a calibrated camera with known position is required and the captured object must have a complex mirror surface. Such a surface has multiple elliptic and hyperbolic regions that reflect the same environmental feature multiple times at multiple surface points. In their experiments they place the mirroring object in a room with a colorful environment and then use RGB-SIFT to establish so-called *static reflection correspondences* (SRCs) between the multiple reflections of unique environmental features. The following reconstruction algorithm assumes that the environment is infinitely far away and the camera can be modeled orthographically. Based on the fact that in an elliptical region with positive Gaussian curvature only one SRC can occur and thus any surface normal is unique, a clustering algorithm is able to separate the surface into multiple either elliptical or hyperbolical patches. The authors model these surface patches then using local quadrics consisting of 5 parameters. A quadric is an implicitly defined surface using a quadratic polynomial. Again, they use the SRCs which occur at different surface patches to set up a constraint that the normal at these SRCs has to be equal in each surface patch. This property relies on the fact that the features are assumed to be infinitely far away. The authors were able to formulate a system of linear equations which they solve using a convex optimization toolset. The normals at several surface points have to be provided to avoid global ambiguities. Their algorithm is capable of reconstructing high-quality results even on real-world scenes.

Closely related to the approach of this thesis is the approach presented by Godard et al. [2015]. The authors present a multi-view technique that is able to reconstruct highly specular surfaces under natural illumination. The main difference to this thesis is, however, that an environment map has to be captured. They reconstruct the specular surface by iteratively estimating normals and updating the surface based on these normals. They initialize the surface by a voxel carving technique resulting in the visual hull of the object using image masks that separate the specular object from the background. The authors create these masks manually for each input image. Correspondences between the environment map and the input images are established by matching color information. However, due to the use of a color-based matching, many ambiguous matches can occur. The authors solve this problem by modeling the normal inference as a probabilistic approach where multiple measurements from different views increase the probability of a correct match. Once the matching between input images and environment map is unambiguous, the normals of the corresponding vertices are updated using the correct or rather most probable reflection direction from the environment map. Then the surface mesh is corrected such that it conforms to the new normals. This process of updating normals and the mesh is iterated until convergence. Their algorithm is able to reconstruct specular objects starting with their visual hull with a low error compared to a laser-scanned ground truth. In addition, they show that an explicit modeling of inter-reflections leads to a further decrease of the reconstruction error.

3 Reconstruction

The main idea of the reconstruction is to use an iterative optimization on a parametric surface. The objective function of this non-linear optimization is the difference between the reflection of corresponding viewing rays and the actual direction of the environmental feature.

3.1 Design Criteria

To approach the primary goal of this thesis—the reconstruction of specular surfaces in uncontrolled environments—a framework was developed allowing to be flexible regarding different types of input data or optimization modelings.

First of all, the framework should be capable of processing different input configurations such as (un)known environmental feature positions, (un)known feature directions or known surface positions. This was mandatory for the experiments because it was unclear if the algorithm would be able to produce meaningful results using the respective input data. This design is also beneficial when extending the framework for a general purpose specular surface reconstruction pipeline.

Secondly, the underlying parametric surface model of the reconstruction should be interchangeable. This makes it possible to choose a more adapted model for the specific surface geometry of the specular object. The two main requirements for the surface model are the ability to perform ray–surface intersections and that the number of parameters is preferably low to allow for a feasible optimization.

The third and last subgoal is the ability to exchange and extend the problem modeling and the formulations of the error functions within the optimization framework.

3.2 Prerequisites and Input Data

The reconstruction algorithm expects multiple high-resolution images from different viewpoints of a specular object as input data. We need to estimate the extrinsic and intrinsic parameters in advance, for instance using an SfM algorithm. Hence the input images are required to contain sufficiently many diffuse, textured objects. The Multi-View Environment [Fuhrmann et al., 2015] is a pipeline for image-based geometry reconstruction and features an SfM application which we use to accomplish this task.

3.2.1 Feature Matching

Finding pixel-to-pixel correspondences for pairs of input images is a significant preprocessing step of the reconstruction algorithm. For instance, we can acquire correspondences using feature matching or by estimating the optical flow. Preliminary experiments showed that extracting long feature tracks from an image series or video did not work reliably. For (dense) optical flow, the reason for this lies in the fact that errors in the pixel-to-pixel motion accumulate over the image series.

Although we have to perform an image feature matching during the camera calibration, the resulting feature matches are not suitable as input to the reconstruction framework. The reason for this is that feature matching for SfM typically uses two tests to filter out wrong matches. The first test, proposed by Lowe [2004], is a ratio threshold on the distance between the first and the second best matching feature descriptor. This is more reliable to separate outliers from valid matches than a simple threshold on the distance of feature descriptors. However, in a scene with a specular object, an environmental feature can get reflected at multiple positions. Multiple features with similar descriptors render the ratio test unsuitable.

The second common test in feature matching is a RANSAC-based algorithm which tries to find the largest set of feature matches that can be explained through a fundamental matrix (**F**) and classifies all other matches as outliers. Again, we cannot apply this test to feature matching of specular objects because we have to rely on epipolar geometry for feature track triangulation. This is not possible because the epipolar constraint does not hold for feature reflections generated by specular objects.

The omission of these tests leads to an intractable number of wrong feature matches. Since we want to use the matches later in a least-squares-based optimization where outliers can lead to problematic results, a small number of wrong matches is needed. We can observe that sharp features often appear inside smaller clusters on the surface. Approximately, we can see these clusters as quasi-planar patches which enables us to use a novel outlier removal strategy, proposed as follows:

1. Mark all matches as putative outliers.
2. Use RANSAC to find a substantially large set of matches that lie in a cluster whose transformation can be explained using a homography matrix (**H**). The matches in

this set can now be considered inliers.

3. Repeat the previous step as long as the number of new inliers is larger than N.

We can see a result of this new algorithm in Section 4.2.2. In this implementation, the parameter configuration $N = 15$ yielded the best matching results.

3.2.2 Specular Feature Tracks

In theory, the specular features should be separable from the diffuse ones because their triangulated 3D position is not unique but rather located on caustic curves [see Swaminathan et al., 2002]. This means, diffuse features move with constant velocity (on the image plane) if the camera velocity is also kept constant. Specular feature tracks, however, alter their speed based on the curvature of the surface (see again Figure 2.4).

Practical experiments using non-rectified camera setup have shown that the discrimination between solid and reflected features based on their velocity and acceleration is often complicated and imprecise because the camera undergoes rotation and non-linear movement.

An alternative is to use a semi-automatic procedure to classify the feature matches. When using a reconstruction pipeline which is only capable of reconstructing diffuse objects, specular objects will be left out of the final mesh or will receive a very low confidence. We can then use a global confidence threshold to clean the result mesh, render it from all views and consider it as a mask for feature classification. Of course, we could also draw these masks manually.

3.3 Ray Tracing-based Surface Reconstruction

3.3.1 Terminology

Although this thesis tries to follow a nomenclature which is common in computer science, a table of notation which is used in this chapter reads as follows:

P :	a point	$f(\cdot)$:	a function
$\mathbf{n}$:	a normalized vector	$B_{i,k}(\cdot)$:	B-spline basis function
$\mathbf{S}$:	a surface or line	Θ :	a set of parameters
a :	a scalar factor	$\cdot_{\Theta}$:	an entity depending on Θ
$\langle\cdot,\cdot\rangle$:	dot product	$\cdot^{i}$ or $\cdot_{i}$:	an index
P_x :	x-component of P		

3.3.2 Objective Function

The idea of the main reconstruction step is to intersect viewing rays against the surface and minimize the angle between the reflection ray and the feature direction.

We can think of the surface $\mathbf{S}_\Theta$ as a set of points $S_\Theta = (x, y, f_\Theta(x, y))^\intercal$ where $f_\Theta : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a function that depends on the parameter set Θ and describes the height z at x and y. Certainly, this parametrization is just a heightfield which is incapable of representing sophisticated surfaces, however, we need a low number of parameters in the context of optimization. We can evaluate the surface normal vector $\mathbf{n}_P$ at point $P = (x, y, z)^\intercal$ for surface parameters x and y using the cross product of the first derivatives with respect to x and y:

$$\mathbf{n}_P = \frac{\frac{\partial \mathbf{S}_\Theta}{\partial x} \times \frac{\partial \mathbf{S}_\Theta}{\partial y}}{\left\| \frac{\partial \mathbf{S}_\Theta}{\partial x} \times \frac{\partial \mathbf{S}_\Theta}{\partial y} \right\|_2} \tag{3.1}$$

For each image feature belonging to a track, we create a ray from the camera center C through the pixel position of the feature. In addition, we have to transform the ray into a global coordinate system. The ray is represented by a set of points starting at C and pointing into the direction $\mathbf{d}$:

$$\mathbf{R} = C + a\mathbf{d} \tag{3.2}$$

Given the intersection of the ray and the surface at point $P = \mathbf{R} \cap \mathbf{S}_\Theta$ and the surface normal $\mathbf{n}_P$, we can compute the direction $\mathbf{r}$ of the reflection vector:

$$\mathbf{r} = \frac{\mathbf{d} - 2\langle \mathbf{d}, \mathbf{n}_P \rangle \mathbf{n}_P}{||\mathbf{d} - 2\langle \mathbf{d}, \mathbf{n}_P \rangle \mathbf{n}_P||_2} \tag{3.3}$$

To begin with, we can assume that the direction of the feature $\mathbf{f}$ is either known or we can calculate it using the known feature position F:

$$\mathbf{f} = \frac{F - P}{||F - P||_2} \tag{3.4}$$

We can then formalize the surface reconstruction as the following minimization problem:

$$\min_\Theta \sum_i \angle\left(\mathbf{r}^i_\Theta, \mathbf{f}^{m(i)}\right)^2 \tag{3.5}$$

Note that the reflection ray direction $\mathbf{r}_\Theta$ depends on the parameters Θ because the intersection point P as well as the normal $\mathbf{n}_P$ depend both on the current shape of the surface which is modeled by Θ. The mapping function $m(\cdot)$ returns the index of the common feature direction for each ray i to express that multiple reflection rays correspond to the same feature direction $\mathbf{f}^{m(i)}$. We can either define the error function $\angle(\cdot,\cdot)$ as the angle between the two directions or the squared sum of differences of spherical angles:

$$\angle(\mathbf{a},\mathbf{b}) = \arccos(\langle\mathbf{a},\mathbf{b}\rangle) \tag{3.6}$$

$$\angle(\mathbf{a},\mathbf{b}) = \|\mathbf{a}_\theta - \mathbf{b}_\theta\|_2^2 + \|\mathbf{a}_\phi - \mathbf{b}_\phi\|_2^2 \tag{3.7}$$

The latter makes it possible to represent the normals by only two parameters, inclination θ and azimuth ϕ, in contrast to x, y and z. We can achieve this lower-dimensional representation by a simple transformation:

$$\theta = \arccos(\mathbf{d}_z) \tag{3.8}$$

$$\phi = \operatorname{atan2}(\mathbf{d}_y, \mathbf{d}_x) \tag{3.9}$$

Note that the atan2(y,x)-function is a convenience wrapper for $\arctan(\frac{y}{x})$ which also defines the cases when some of the input arguments are zero.

3.3.3 Unknown Feature Direction

Equation (3.5) still assumes a known feature direction $\mathbf{f}$. A straight forward way to remove this assumption in the optimization problem is to jointly optimize the feature direction $\mathbf{f}$ together with Θ:

$$\min_{\Theta,\mathbf{f}} \sum_i \angle\left(\mathbf{r}^i_\Theta, \mathbf{f}^{m(i)}\right)^2 \tag{3.10}$$

To decrease the number of optimization variables, we can represent the feature directions using the spherical coordinate system as we have seen in Equations (3.8) and (3.9). Another possible adaption of the optimization problem is to build the residuals in form of pairs of rays which see the same feature and thus should point into the same direction:

$$\min_{\Theta} \sum_{\substack{i,j \\ m(i)=m(j)}} \angle\left(\mathbf{r}^i_\Theta, \mathbf{r}^j_\Theta\right)^2 \tag{3.11}$$

This approach retains the number of optimization variables but increases the number of residuals quadratically in the number of rays.

As mentioned earlier, the optimization is prone to converge to local optima. To overcome this issue, we can introduce a second residual function. This residual enforces the surface patch to be fitted through known surface points, e.g. at the boundary. We can do this before the actual optimization using reflection error residuals or in a combined optimization following both goals. We can define the error as the distance from the surface point to the projection point on the surface. This leads to the following optimization problem:

$$\min_{\Theta} \sum_{s} \left\| f_{\Theta}(B^s_x, B^s_y) - B^s_z \right\|_2^2 \tag{3.12}$$

All mentioned residual functions were modeled using the Ceres Solver [Agarwal et al.], a C++ library designated for solving large and difficult non-linear least squares optimization problems. It is not possible to use the included automatic differentiation techniques because of the ray–surface intersection which we will see in the next section. Thus, we have to evaluate gradients using numeric differentiation. Note that automatic differentiation is not the same as numerical differentiation and requires the use of dual numbers in the implementation of residuals.

3.3.4 B-Spline Heightfield

For the sake of simplicity we can use a 2.5D heightfield as opposed to a full 3D parametric surface. This drastically reduces the number of parameters in the optimization problem. The number of possible surfaces which can be represented by such a heightfield is lower compared to the full surface because the height values are represented by a function.

In this specific implementation the surface of the specular object is modeled as a B-spline heightfield

$$\mathbf{S}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} h_{i,j} B_{i,k}(u) B_{j,l}(v) \tag{3.13}$$

with the knot vectors $\mathbf{u} = (u_0, u_1, \cdots, u_{m+k})$ and $\mathbf{v} = (v_0, v_1, \cdots, v_{n+l})$ and a rectangular set of control height points $h_{i,j}$ where $0 \leq i \leq m$ and $0 \leq j \leq n$. In this implementation cardinal cubic B-spline surfaces were used, i.e. the knot vectors were evenly spaced and the degree was set to $k = l = 3$.

The basis functions are defined recursively:

$$B_{i,1}(x) := \begin{cases} 1 & \text{if } u_i \leq x \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

$$B_{i,k}(x) := \frac{x - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(x) + \frac{u_{i+k} - x}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(x) \tag{3.15}$$

We can calculate the derivative of the B-Spline basis function using the following scheme:

$$\frac{dB_{i,k}(u)}{du} = (k-1) \left(\frac{-B_{i+1,k-1}(u)}{u_{i+k} - u_{i+1}} + \frac{B_{i,k-1}(u)}{u_{i+k-1} - u_i} \right) \tag{3.16}$$

The main reason to use a heightfield as opposed to a 3D surface is the ability to calculate ray–surface intersections efficiently and in a simple fashion. If the viewing ray **R** is in the same coordinate system as the surface **S**, we can find the intersection point using a binary search along the ray by determining the scalar a such that the following equations are fulfilled:

$$P := a\mathbf{r} + C = (P_x, P_y, P_z)^\intercal \tag{3.17}$$

$$\mathbf{S}(P_x, P_y) = P_z \tag{3.18}$$

We can find the infimum and supremum of a using an intersection test with the ray against the bounding box that contains the control points. Note that by the property of B-splines the bounding box of the control points is also a bounding box for the entire surface because the control points define the convex hull of the surface.

3.3.5 Initialization of the Iterative Optimization

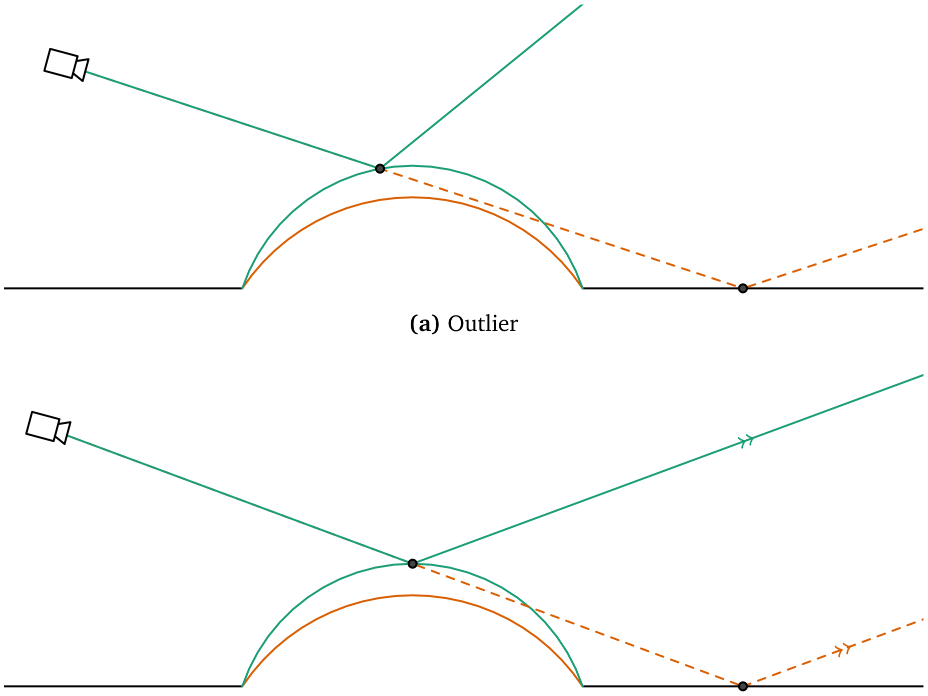
We can summarize the initialization of the iterative optimization as the following steps: First, we fit a plane to the boundary points. We can then use this plane to initialize the parametrization of the B-Spline surface. To define the dimensions of the patch, we project the surface points onto this plane and acquire the bounding rectangle. We use the coordinate system that is defined by this rectangle in the following. Now, we sample m times n points on the rectangle and use them as initial control points $h_{i,j}$ of the B-spline heightfield. Additionally, we have to transform the viewing rays, surface points and, if available, feature positions into the surface-local coordinate system.

3.4 Robustness Extensions

By adding the feature direction to the set of optimization variables, the problem gets prone to multiple local optima. Even using multiple camera positions, the problem can still be challenging without further treatment. Therefore we need to use several extensions to robustify the optimization process.

3.4.1 Robust Loss Function

Typically, one uses robust loss functions to accommodate for outliers in input data. It is very advisable to use them instead of a simple squared loss when processing real-world input data. In this specific problem formulation, however, even synthetically generated



(a) Outlier

(b) Ambiguity

Figure 3.1: In this example, a viewing ray hits the true surface (green) at the correct intersection point. The intersection point at the slightly incorrect surface (orange dashed) lies substantially farther away from the desired region. This shows that small changes in the surface shape can lead to large jumps of the intersection point. However, the measured error is only dependent on the reflection vector direction. This direction can either be different (outlier) or the same (ambiguity) as the true reflection direction.

outlier-free input data can lead to outliers in terms of the residual function. We can see the impact of these outliers on the reconstruction results later in Chapter 4 “Experiments and Results”.

The reason for this behavior lies in the strong non-linearity of the error function: Even small changes in the surface parameters can lead to jumps of the residual. For example, if the viewing ray hits the surface at a grazing angle, a small change of one or more nearby control points can move the intersection point a substantially large step away from the previous position (see Figure 3.1). Since the residual function depends only on the reflection direction instead of the intersection point position, again ambiguities can arise. A robust loss function can only remove outliers in terms of a wrong intersection point if this outlier also produces an incorrect reflection direction.

Therefore, the framework is capable of using the robust loss functions which are provided by the Ceres Solver optimization framework. We can see a plot of these loss func-

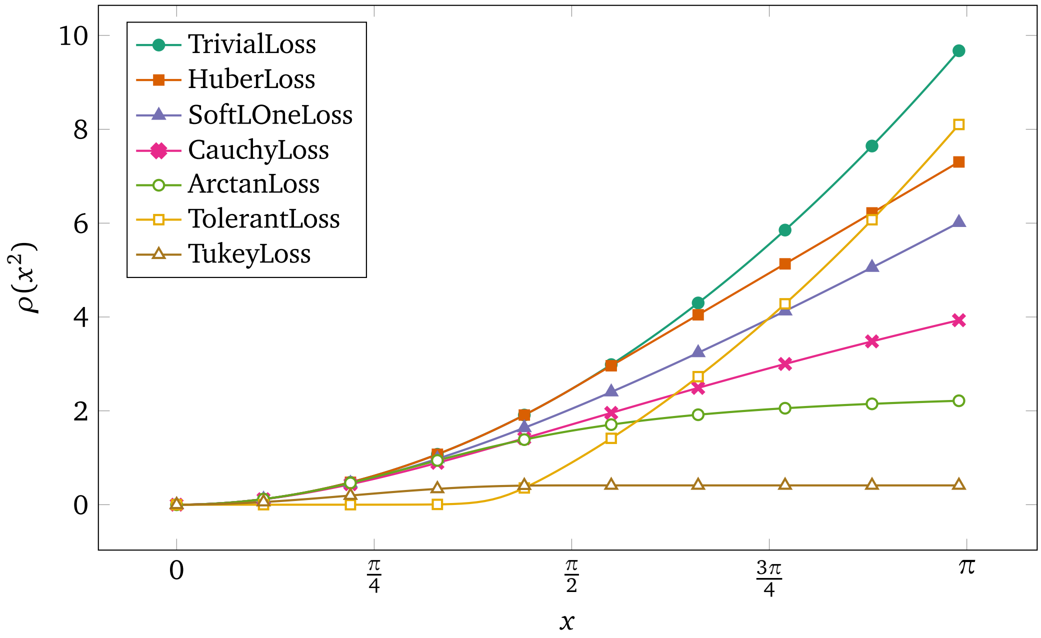


Figure 3.2: Plot of the provided loss functions from the Ceres Solver library in the range of $x \in [0, \pi]$. In this case, x are the angular differences between the reflection rays.

tions in Figure 3.2. We can then update the optimization problem from Equations (3.10) and (3.11) as follows:

$$\min_{\Theta,\mathbf{f}} \sum_{i} \rho_i \left(\angle \left(\mathbf{r}^i_\Theta, \mathbf{f}^{m(i)} \right)^2 \right) \tag{3.19}$$

$$\min_{\Theta} \sum_{\substack{i,j \\ m(i)=m(j)}} \rho_{i,j} \left(\angle \left(\mathbf{r}^i_\Theta, \mathbf{r}^j_\Theta \right)^2 \right) \tag{3.20}$$

In consequence of the results shown in Section 4.1.2.4, the arctan-loss function was used for most of the experiments in this thesis.

3.4.2 Sequential Fitting and Ray Tracing

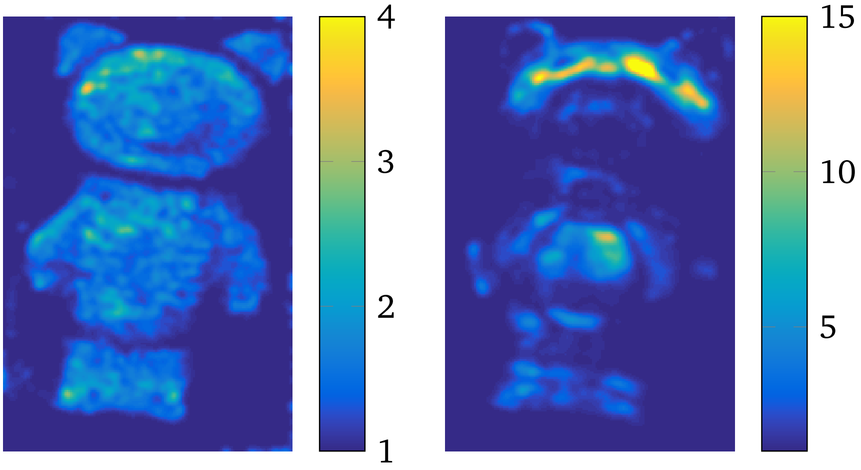
Optimizing two different error functions, such as the fitting error (Equation (3.12)) and the reflection error (Equations (3.10) and (3.11)), simultaneously can be difficult because one has to choose a weighting factor between the two terms. If the weight for one of the error functions is too high, the optimization will neglect the other function.

To circumvent this problem, we can separate the two different error functions and first fit the B-Spline patch to the given points on the surface and thereafter use the reflection error function. In addition, it can be advantageous to fix the control points at the boundary of the patch after the fitting. This assumes that most of the given surface points will be placed at the boundary of the specular object which is typically the case. The disadvantage of this approach is that the solution can drift away from the initialization during the optimization.

3.4.3 Iterative Re-Weighting

In an uncontrolled environment, we cannot assume that the features are uniformly distributed. In practice, there are regions on the specular surface where the number of viewing rays is substantially larger than in other regions. We can see a comparison between uniformly distributed synthetic features and real-world environments in Figure 3.3. As we can later see in Chapter 4 “Experiments and Results”, this leads to distorted surface parts in regions with low density of ray intersections. To fix this issue, we can scale each residual with a specific weight w_i. This weight is the inverse density of rays that hit the surface.

We have to calculate the new weights between each of the iterations because the location of intersection points change in each optimization iteration. Considering the formulation, we can apply the weights as a scaling factor for the output of the loss



(a) Synthetic input data

(b) Real-world input data

Figure 3.3: The number of rays hitting the surface patch. In the synthetic case, the distribution is more uniform compared to real-world input data. We use the inverse of this 2D histogram later to re-weight the residuals corresponding to the viewing ray.

function ρ:

$$\min_{\Theta,\mathbf{f}} \sum_i w_i \rho_i \left(\angle \left(\mathbf{r}^i_\Theta, \mathbf{f}^{m(i)} \right)^2 \right) \tag{3.21}$$

$$\min_{\Theta} \sum_{\substack{i,j \\ m(i)=m(j)}} w_{i,j} \rho_{i,j} \left(\angle \left(\mathbf{r}^i_\Theta, \mathbf{r}^j_\Theta \right)^2 \right) \tag{3.22}$$

To compute the ray density, we discretize the surface using a histogram that has a width and height ten times the number of control points in x and y direction. Each histogram bin counts the number of samples hitting the surface at the corresponding region. We can compute this histogram using the same intersection test as described above. After counting all samples, we blur the histogram using a Gaussian filter to smooth the weights. We have to chose the filter parameter depending on the density of ray-intersections on the surface. For each pixel, we use the inverse of the count value as a weight for each residual. Note that we have to initialize the histogram with ones in order to be able to compute the inverse.

A property of this approach is that residuals at regions with a low density receive a higher weight. This can be counterintuitive because it basically amplifies the influence of outliers if they occur in regions of low density. This is a drawback of the proposed approach and requires input data with as few outliers as possible.

4 Experiments and Results

In order to show the feasibility of the proposed reconstruction technique, synthetic as well as real-world test scenarios were used. First, synthetic tests were created to avoid problematic outliers or measurement errors. These synthetic scenes were modeled using the 3D graphics software package Blender [Blender Foundation]. This application enables the easy creation of surfaces, surrounding features and virtual cameras.

Subsequently to the synthetic tests, real-world datasets were captured to evaluate the algorithm on realistic input data.

4.1 Synthetic Dataset

A simple synthetic dataset based on a B-spline surfaces consisting of a regular 7×7 grid of control height points was created by first setting all points to equal height. Then one of the points was shifted slightly above the plane to create a convex bump, another was moved downwards to create a concave dent (later referenced as CONVEXCONCAVE dataset). The experiments used a planar regular grid with the same proportions and number of parameters as initialization for the optimization.

Environmental feature locations on a hemisphere above the surface patch were generated using rejection sampling. Using this approach, a variable number of features at a variable distance could be sampled. Because the hemisphere as well as the surface were centered at the origin, it was also possible to interpret the feature positions as directions for certain experiments. In addition to the features, 25 virtual cameras were placed 2.5 world units above the surface which had an edge length of 1.8 world units. The look-at-point was always the center of the surface. We can see an example synthetic scene configuration using features with a distance of 10.0 world units to the origin in Figure 4.1.

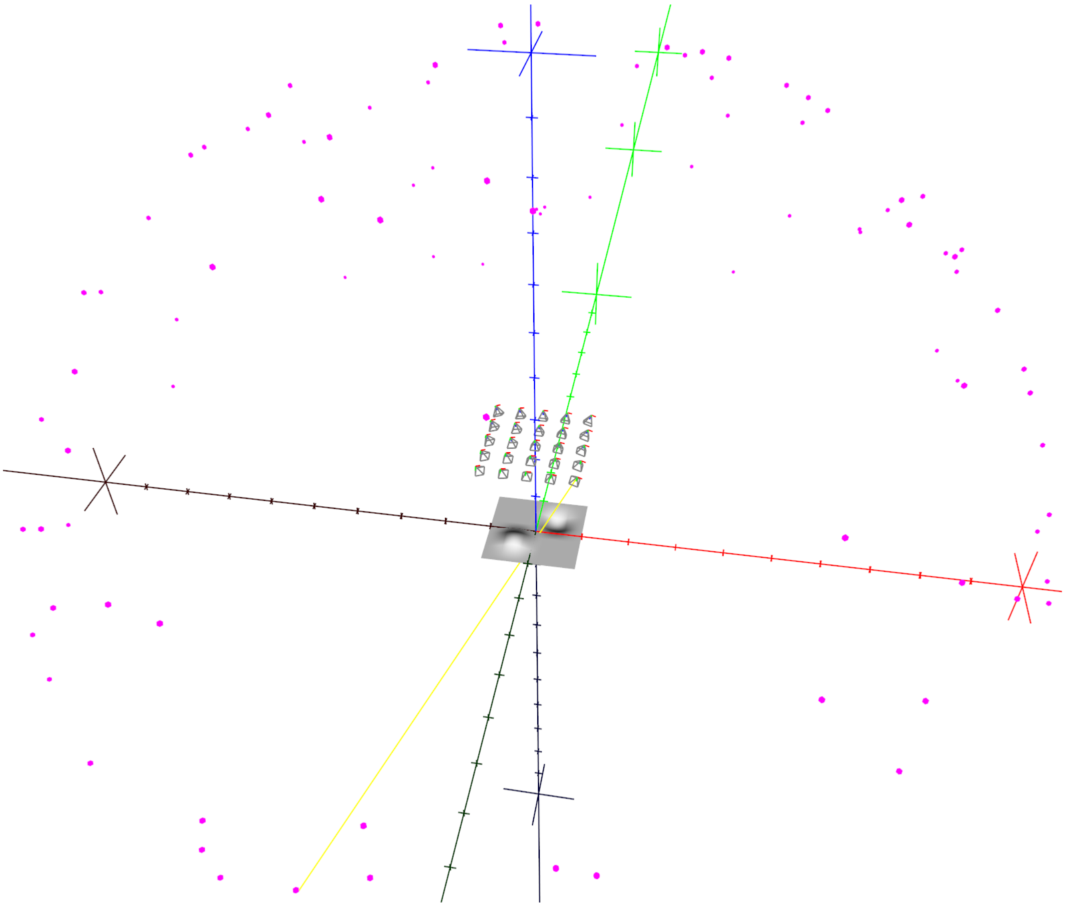


Figure 4.1: Rendering of the ConvexConcave synthetic test scene. Above the shown ground truth surface 25 camera frusta can be observed. The pink dots are unique environmental features distributed on a hemisphere above the surface.

4.1.1 Obtaining Reflectance Correspondences

Instead of rendering the synthetic scene and using image feature matching, the correspondences were acquired using a simple ray tracing approach. The image plane of the virtual cameras were discretized using 1000 × 1000 pixels. For each pixel a ray was traced through the scene. Then the distances of the environmental feature points to the reflection of the ray from the surface were measured. If this distance was reasonably small, the reflection ray was considered to see the environmental feature and the ID of the feature was stored for this pixel. To simulate features at infinity the angular distance between the reflection ray and the feature direction was used to evaluate whether the reflection ray was seeing the feature. Using such an approach, nearby pixels can be assigned the same feature ID which leads to unnecessary correspondences. These

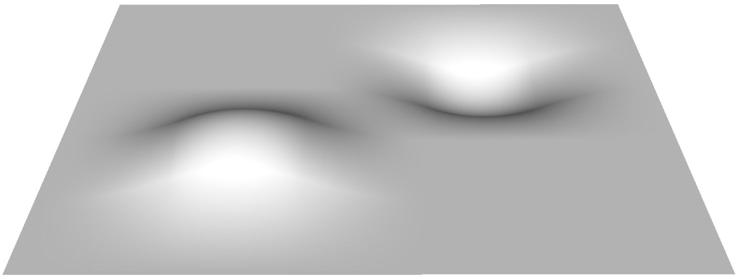


Figure 4.2: Rendering of the reconstructed synthetic surface patch of the ConvexConcave dataset using known feature positions. There is no noticeable difference to the ground truth surface.

additional pixels can be removed using a standard non-maximum suppression based on the inverse of the feature-to-ray distance. The pixels with identical feature ID were then combined to a feature track.

4.1.2 Reconstruction

4.1.2.1 Known Feature Positions

To ascertain that the proposed reconstruction technique is able to reconstruct specular surfaces at least using constrained input data with known feature positions, the algorithm was tested on the presented synthetic test scene. The result of this basic test scenario can be seen in Figure 4.2. This result complies with the results from recent publications (see Chapter 2) in the fact that the reconstruction using known feature positions and multiple cameras can be done accurately. Note that for this and subsequent experiments, the control points at the boundary were fixed during the optimization (see also Section 3.4.2).

The following sections show a more quantitative evaluation of the reconstruction using varying input parameters. For example, the distance of the features and the required number is examined.

4.1.2.2 Unknown Feature Positions

Since the reconstruction in uncontrolled environments is the primary goal of this thesis, the following experiments were executed without the knowledge of the environmental feature positions. For the test on the ConvexConcave dataset the residual functions without the weighting (Equations (3.19) and (3.20)) were used because the environmental features were uniformly distributed.

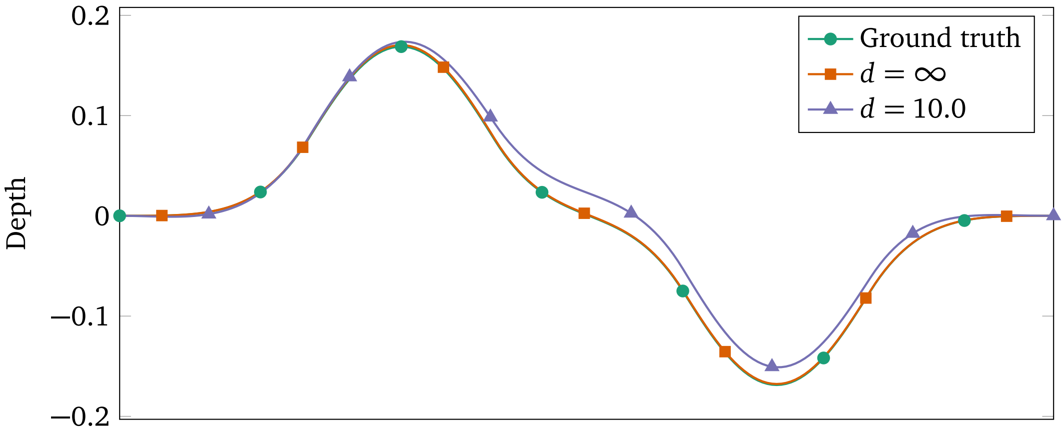


Figure 4.3: The plotted lines represent a diagonal cut trough the reconstructed surface of the Convex-Concave dataset using unknown feature positions. The ground truth surface is hardly visible in the plot because it overlaps with the reconstruction using infinitely distant features ($d = \infty$).

Figure 4.3 shows that the reconstructed patch using features at infinity is almost identical to the ground truth. However, the same features with a fixed distance of 10.0 world units to the surface lead to a slightly erroneous surface. We can also see that the error in the concave area is slightly higher than in the convex area. A possible explanation for this result is that the number of reflections is typically higher in convex surface region leading to a more accurate result. Convex regions reflect only a smaller part of the surrounding environment into the camera. Quantitative errors were calculated by converting the B-spline surface to a triangle mesh and computing the mean difference in height to the ground truth vertices (hereinafter referred to as *mean mesh error*). While the reconstruction using infinitely distant features resulted in a mesh error of 7.90×10^{-4}, the nearby features yielded an error of 1.13×10^{-2}.

4.1.2.3 Distance of Features

The short distance of the environmental features to the surface appears to be the main cause for the previously shown erroneous reconstructed surface. It should be remembered that the algorithm assumes the features to be infinitely far away. The influence of this distance was analyzed in another experiment. Using the same scene and parameters from the previous experiments, a series of measurements were created by increasing the distance 10 and 100 times compared to the original distance. In addition, a measurement using infinitely far away features was computed.

The result of this series can be seen in Figure 4.4 where the mesh error decreases with increasing distance. The largest tested distance yields an almost similar result in

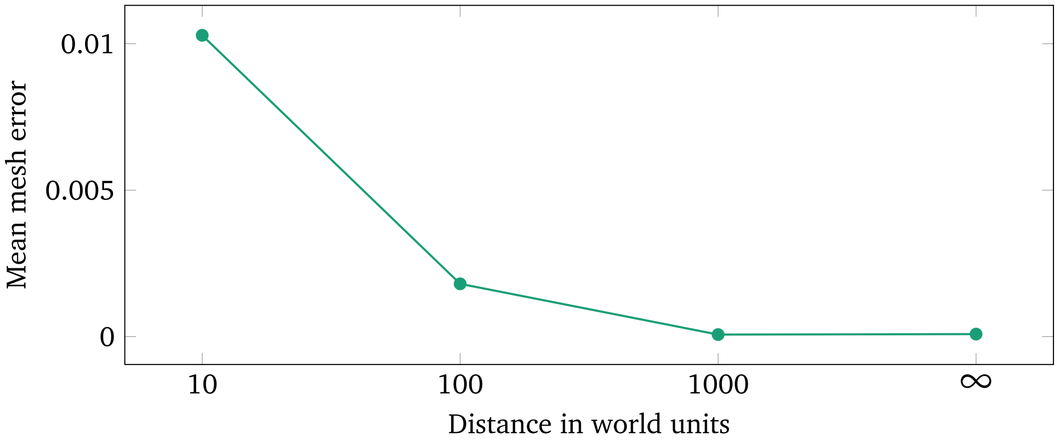


Figure 4.4: The plot shows the mean mesh error of the reconstruction compared to the ground truth. As expected, the error decreases with the distance. With a distance of 1000.0 world units, the error is as low as using features at infinity.

comparison to the reconstruction using infinitely distant features. The reason for this is that the direction vector from each point of the surface towards the feature position becomes almost identical with increasing feature distance.

4.1.2.4 Robustness Functions and Convergence

Section 3.4.1 pointed out that robust loss functions are needed to achieve accurate reconstructions even using noise-free input data. However, there are numerous loss functions that can be used. The Ceres Solver library provides a set of loss functions which can be easily exchanged. In Figure 4.5 the error is displayed in the course of the iterative optimization. We can observe that there are only two loss functions where the optimization converges to an optimal solution, namely the TukeyLoss and the ArctanLoss. Both functions converge to a constant asymptotically. This has the effect that only residuals which are close to the optimum have an impact on the numerical derivatives. Changes in residuals with a large error are almost ignored in the optimization.

4.1.2.5 Number of Features

One desired property of reconstruction algorithms is the ability to process sparse input data. To evaluate the algorithm's performance regarding this property, the influence of the number of environmental features on the mesh error was analyzed. Features at infinity as well as nearby features were evaluated for both proposed residual functions (Equations (3.19) and (3.20)) resulting in four measurement series, shown in Figure 4.6.

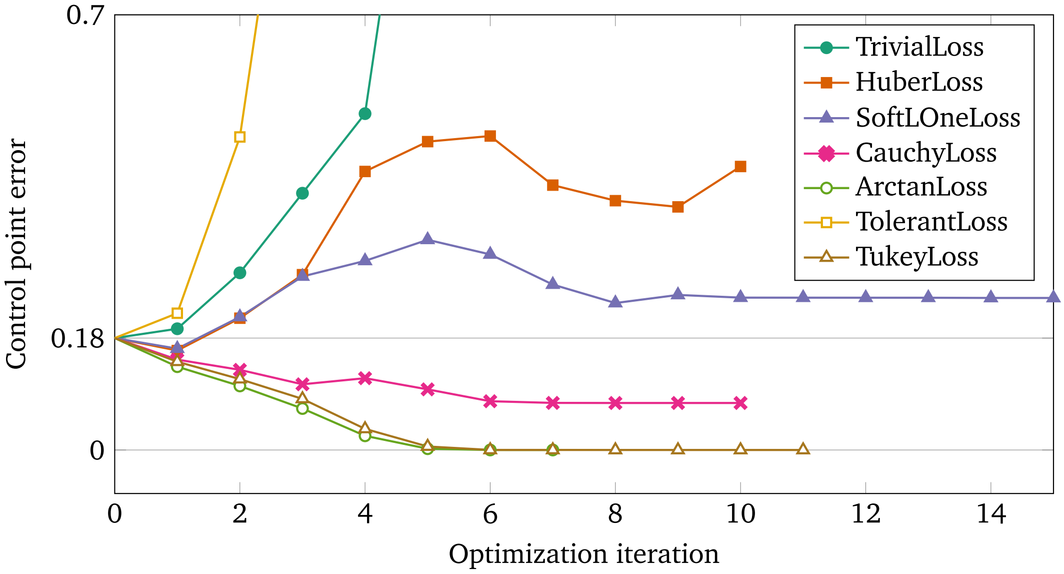


Figure 4.5: This is a plot of the resulting error of the current surface to the ground truth using different loss functions. The error is plotted for each iteration until the optimizer converged or stopped otherwise. For implementation reasons, the sum of squared distances from the current control points to the ground truth control points was used. Note that an error of 0.18 corresponds to the plane which was used as initialization. Errors above that value indicate that the optimization led to a degenerated surface.

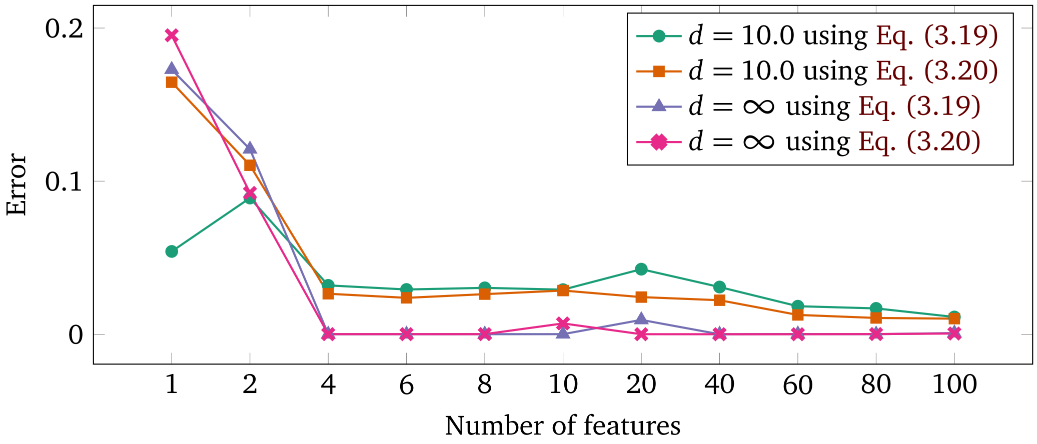
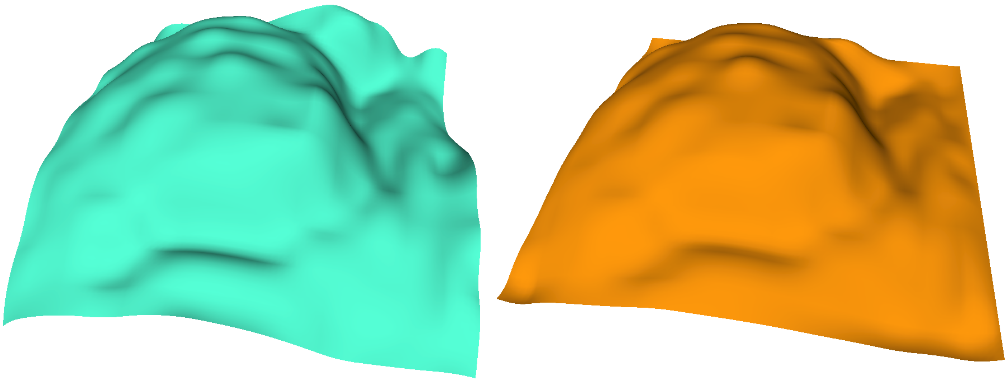


Figure 4.6: This shows a plot of the reconstruction error with different number of environmental features. The experiment was executed using both proposed residuals presented in Equations (3.19) and (3.20). In addition, the feature distance was set to $d = 10.0$ and $d = \infty$, respectively.



(a) Ground truth

(b) Reconstruction

Figure 4.7: *Left:* A B-spline surface patch which was fitted to a part of the Stanford bunny to serve as ground truth. *Right:* The reconstruction of this complex specular object is reconstructed accurately in the middle of the patch. The error at the border is higher because the outer control points were fixed after initializing the B-spline surface with a plane.

For this specific dataset, as few as four environmental points are sufficient to achieve an accurate reconstruction. In general, it can be observed that a larger number of features leads to a more accurate reconstruction, especially when the features are closer to the surface. Both residual formulations perform equally using $d = \infty$. With $d = 10.0$, the second formulation (Equation (3.20)) yields a slightly lower error.

4.1.3 Complex Specular Objects

The synthetic surface used so far is very simple and was reconstructed with low errors. As a more challenging synthetic scene, a part of the Stanford bunny was used for the next dataset. A B-spline patch using 15 × 15 parameters was fitted against the original mesh and used as ground truth (hereinafter named BUNNY dataset). The camera and feature positions were constituted like in the previous experiments. In Figure 4.7 we can see that the reconstruction works well also on more complex objects. In this experiment the reconstruction was initialized with a plane.

4.2 Real-World Datasets

The previous experiments on synthetic datasets demonstrated the performance of the proposed algorithm on generated scenes where the distribution of environmental features was controlled. In addition, the matching was simulated leading to exact and

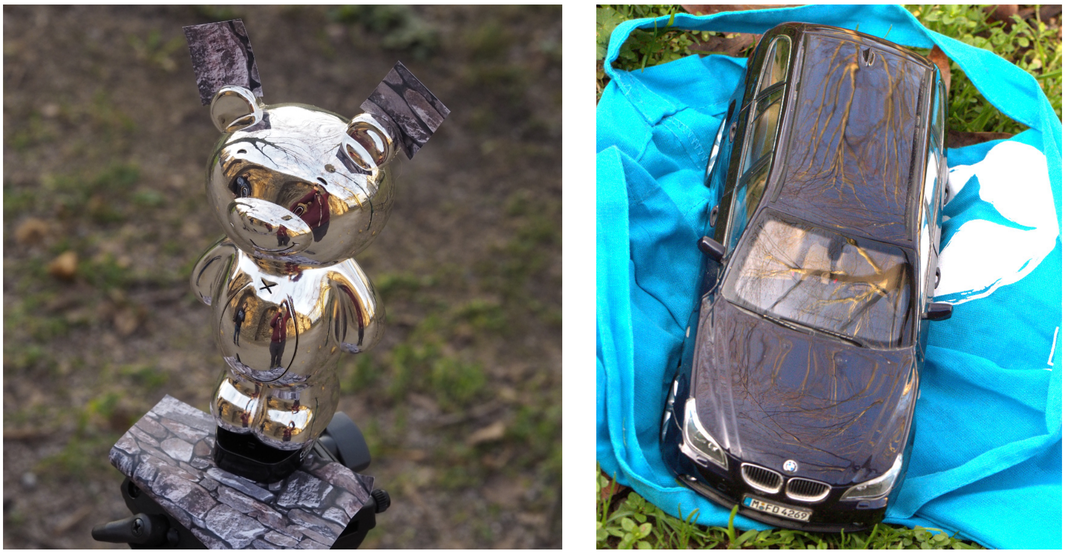


Figure 4.8: *Left:* A photograph of the Teddy dataset which consists of a highly specular surface with only little structure. Paper strips were attached to the object to provide enough diffuse features for SfM. *Right:* The ModelCar dataset has more structure on it but consists of parts with lower surface curvature.

outlier-free correspondences with long feature tracks. Experiments in real-world scenes however, are more challenging. The following sections describe the capture and performance of real-world input data.

Two specular objects were used for evaluation. A complex mirroring surface was provided by the Teddy object. Since there is very little structure on the object itself, a laser-scan of approximately the same object was used from the supplemental material from a similar project [Godard et al., 2015]. The second dataset, namely the ModelCar dataset, is a model car with a specular coating which also has some structure from corners and edges of the single building components. An image of both datasets can be seen in Figure 4.8.

4.2.1 Acquisition

The capture parameters of the presented objects have to be set deliberately. A video using a standard 1080p resolution leads to a narrow baseline image series which is good for tracking specular features. Preliminary experiments showed that the resolution is still not high enough and the compression artifacts from consumer cameras complicate the feature matching. The datasets were captured at 16 and 24 million pixels (depending on the camera) while moving the camera only little between the frames to achieve a narrow baseline. In addition, the aperture was set to a high f-number to increase the captured depth of field. All images were processed at the full dynamic range of the camera. For the TEDDY dataset, small feature-rich paper strips were attached to the object to estimate the camera poses as well as an initial depth and bounding box.

4.2.2 Matching and Classification

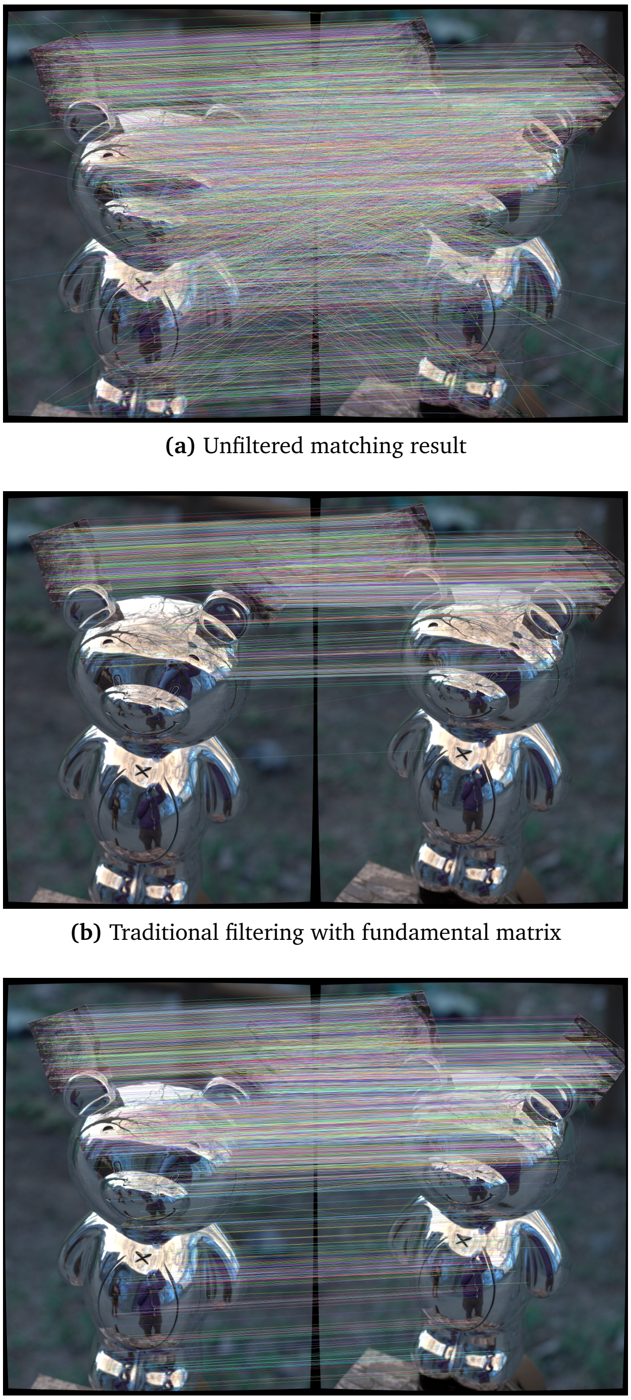
Feature matching was done on both datasets using the proposed matching algorithm. The results from Figure 4.9 show that the adapted matching is able to produce outlier-free feature matches in situations where traditional approaches fail. In addition, another image series of the MODELCAR dataset was taken where two features were tracked manually over the whole surface. The reason for this was the observation that the feature matching did not perform very well on the only slightly curved surface parts of the model car. These surface parts lead to an increased distortion of the image features and require narrow-baseline input images.

The discrimination between diffuse and specular features was accomplished using the proposed ray-tracing approach. Multi-View-Stereo reconstruction followed by a Poisson surface extraction [Kazhdan and Hoppe, 2013] was able to reconstruct a mesh of the edges and corners of the MODELCAR dataset (see Figure 4.10). The laser-scan of the TEDDY dataset was manually aligned to the estimated camera position and the textured paper strips. The scan was then be used as an initialization to the optimization by down-sampling the mesh to a lower resolution or by fitting a B-spline surface with a very low number of parameters to the scan.

4.2.3 Reconstruction

4.2.3.1 ModelCar Dataset

The MODELCAR dataset looked appropriate for an image-based reconstruction because of two reasons: There is certain structure around the specular surface parts and the curvature is rather low. The latter makes it possible to use a parametric surface with a low number of control points which still can lead to good reconstruction results. However, the feature matching did not perform well on the input images resulting in too few correspondences. The track length was in most cases less than four features. There are two main reasons that could explain this observation. First, the material of the model car is not highly specular and possibly consists of a inhomogeneous varnish layer. This results in distorted features and altered feature descriptors leading to less feature matches and a higher amount of outliers. Secondly, with a lower surface curvature, a smaller fraction



(a) Unfiltered matching result

(b) Traditional filtering with fundamental matrix

(c) Adapted filtering for specular objects

Figure 4.9: *Top:* Unfiltered result after feature matching contains a large amount of outliers. *Middle:* Traditional RANSAC-based outlier removal filters away also correct reflected features. *Bottom:* The result of the adapted matching strategy. The number of matches is increased by finding small nearly-planar clusters of features.

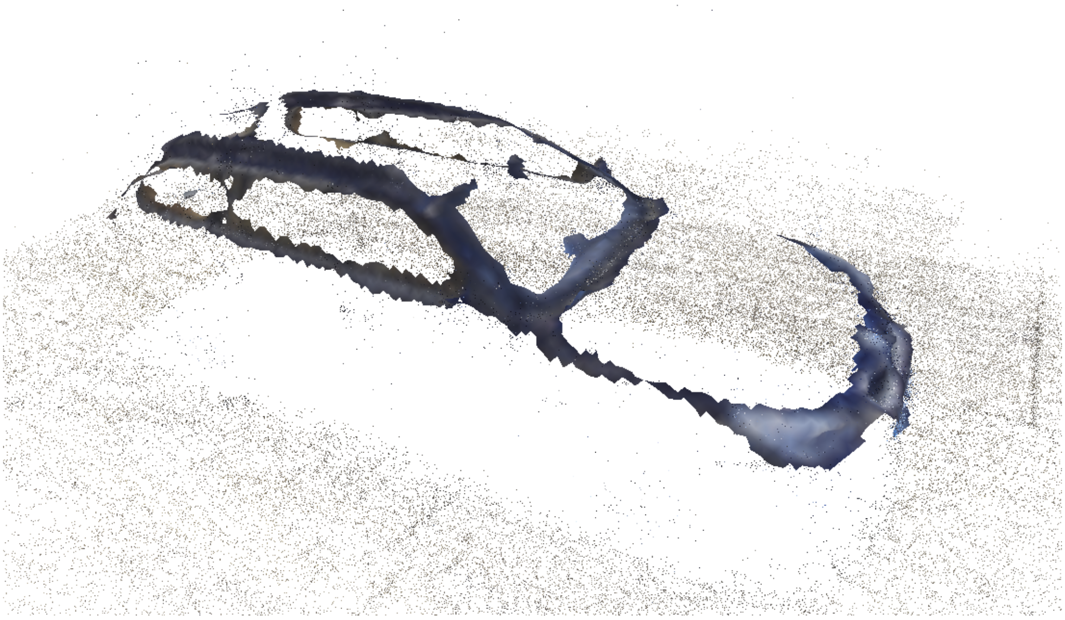


Figure 4.10: The MVS-based reconstruction of the ModelCar dataset. The mesh was cleaned using a confidence threshold to keep only the edges. Reconstructed points from the SfM reconstruction are shown as well.

of the environment is seen by one image compared to surfaces with higher curvatures. This directly leads to a significant decrease of features in each image.

To circumvent these issues, features were tracked manually by annotating a couple of features throughout the image series. This process did not take longer than roughly 30 minutes. Of course, no outlier removal and feature classification has to be performed on the hand-crafted feature matching. A B-spline patch consisting of 8 × 16 parameters was fitted to the already reconstructed edges of the car model (see Figure 4.10).

The reconstruction with unknown environmental features based on this initialization leads to a tilted surface (see Figure 4.11). The reason for this defect is likely an error in the position estimates of the two environmental features. The figure also shows a reconstruction using known feature directions. These directions were estimated manually. This constrains the optimization in such a way that the resulting surface looks similar to the expected surface. A third reconstruction shows a surface using feature positions which were estimated by averaging the reflection rays from the initialization surface. It can be observed that this last result is considerably closer to the surface with manually supplied positions.

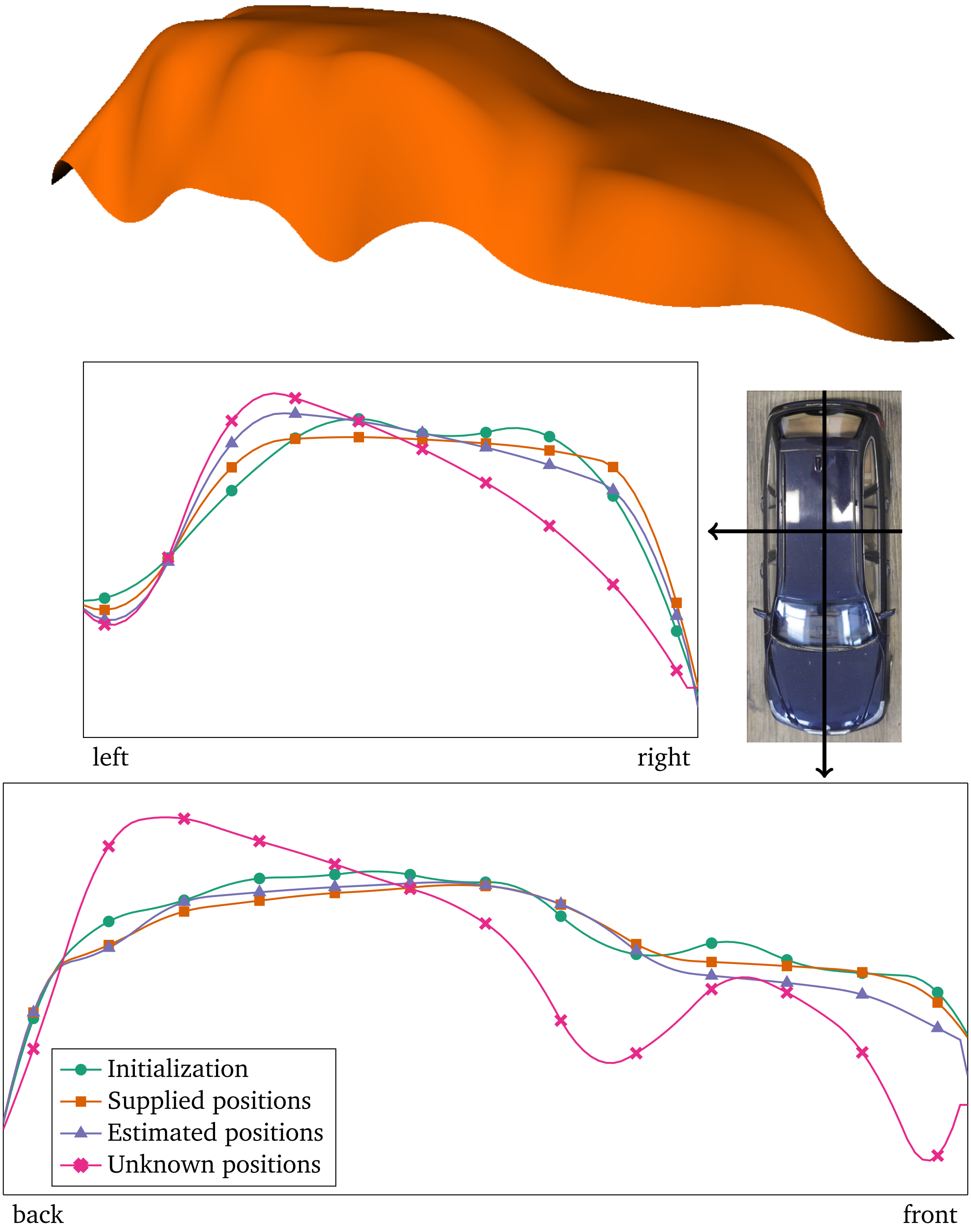


Figure 4.11: On the *top*, a reconstruction of the ModelCar dataset using two manually supplied feature positions is shown. The plots show a horizontal (*middle*) and vertical (*bottom*) cut through different types of reconstructed surfaces as well as the initialization fitted against the reconstructed edges.

4.2.3.2 Feature Distribution of Teddy Dataset

While performing preliminary tests on the TEDDY dataset, the following observation was made: The distribution of feature directions from the input images was not uniform. The visible structure was higher on the surface parts where the surface normal pointed into the upper hemisphere of the surrounding environment. In these preliminary tests, this led to a distorted geometry after reconstruction.

To investigate this observation, a synthetic dataset (hereinafter named SYNTHETIC-TEDDY) was created using the laser-scan of the TEDDY dataset and the same camera positions. Two reconstructions were performed: In the first reconstruction, the features were uniformly distributed whereas in the second test, the number of features in the upper hemisphere was ten times higher than in the lower one. In addition, a reconstruction using the latter feature distribution was computed. This time, the proposed re-weighting of residuals was used (see Figure 4.12). It can be seen in the plots that the residual re-weighting improves the reconstruction but does not remove the effect completely. In addition, it was necessary to start the optimization with a very accurate initialization. Synthetic experiments showed that using a plane or a cylinder as initialization the optimization converged to an incorrect surface.

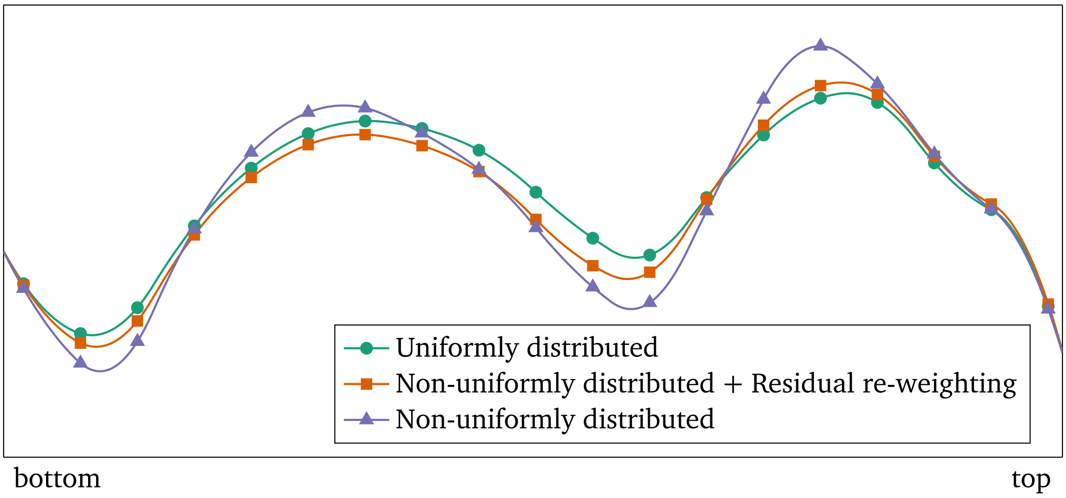


Figure 4.12: A vertical cut through different reconstructions of the SYNTHETICTEDDY dataset. Non-uniformly distributed features lead to a distorted surface reconstruction. The proposed residual re-weighting improves the reconstruction towards the uniformly distributed case.

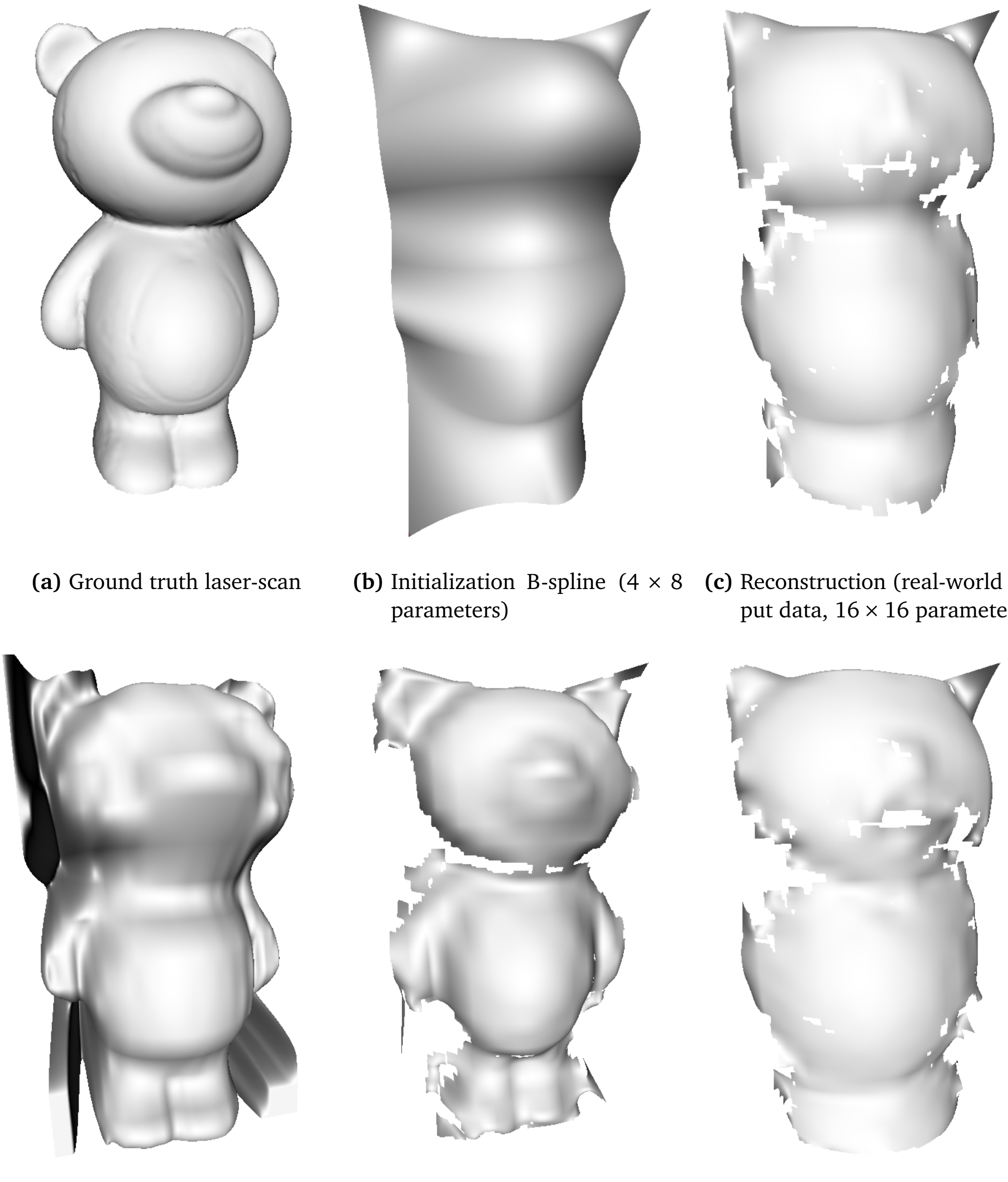
4.2.3.3 Results from Teddy Dataset

For the actual reconstruction experiment of the Teddy dataset, an initialization surface was created by fitting a B-spline patch consisting of 4×8 parameters against the laser-scanned mesh. Thereafter, the reconstruction was performed using 8×16, 16×16 and 16×32 parameters. For the 16×32 case, a reconstruction of the SyntheticTeddy dataset was added to the results for comparison. Figure 4.13 shows renderings of the reconstructed surfaces. First of all, we can see that the resulting surface contains more detail but also visible defects. The nose of the teddy is more detailed in the renderings of the reconstruction in contrast to the initialization patch. A vertical cut through the whole body and a horizontal cut through the nose can be seen in Figure 4.14. This figure shows that the reconstruction depending on the number of surface parameters drifts away from the initialization. This is due to the fact that the fitting to the initialization surface is done before the reflection based optimization. As noted before, a joint optimization would require manually tweaked objective function weights for the fitting and reflection term.

The reconstruction using the synthetic input data (SyntheticTeddy) shows that the proposed algorithm is able to produce results with significant detail. The result has a better visual appearance. The cause for this is likely the uniform distribution of environmental features. The specific distribution of reflections from the real-world input images which was already shown in Figure 3.3 (in Section 3.4.3) leads to specific local optima of the optimization problem.

4.3 Runtime

The proposed optimization problem requires a huge amount of ray–surface intersection tests during the optimization. The intersection tests are automatically parallelized using the Ceres Solver framework. However, the runtime increases dramatically with the number of surface parameters and the number of used features. The experiments were performed on a compute node with two Intel®Xeon®CPU E5-2650 v2 processors. The required amount of memory is low because the algorithm only needs to store the rays, feature positions and the parametric surface; it is mainly compute bound. The runtime of the algorithm for the presented datasets is shown in Figure 4.15.



(a) Ground truth laser-scan

(b) Initialization B-spline (4 × 8 parameters)

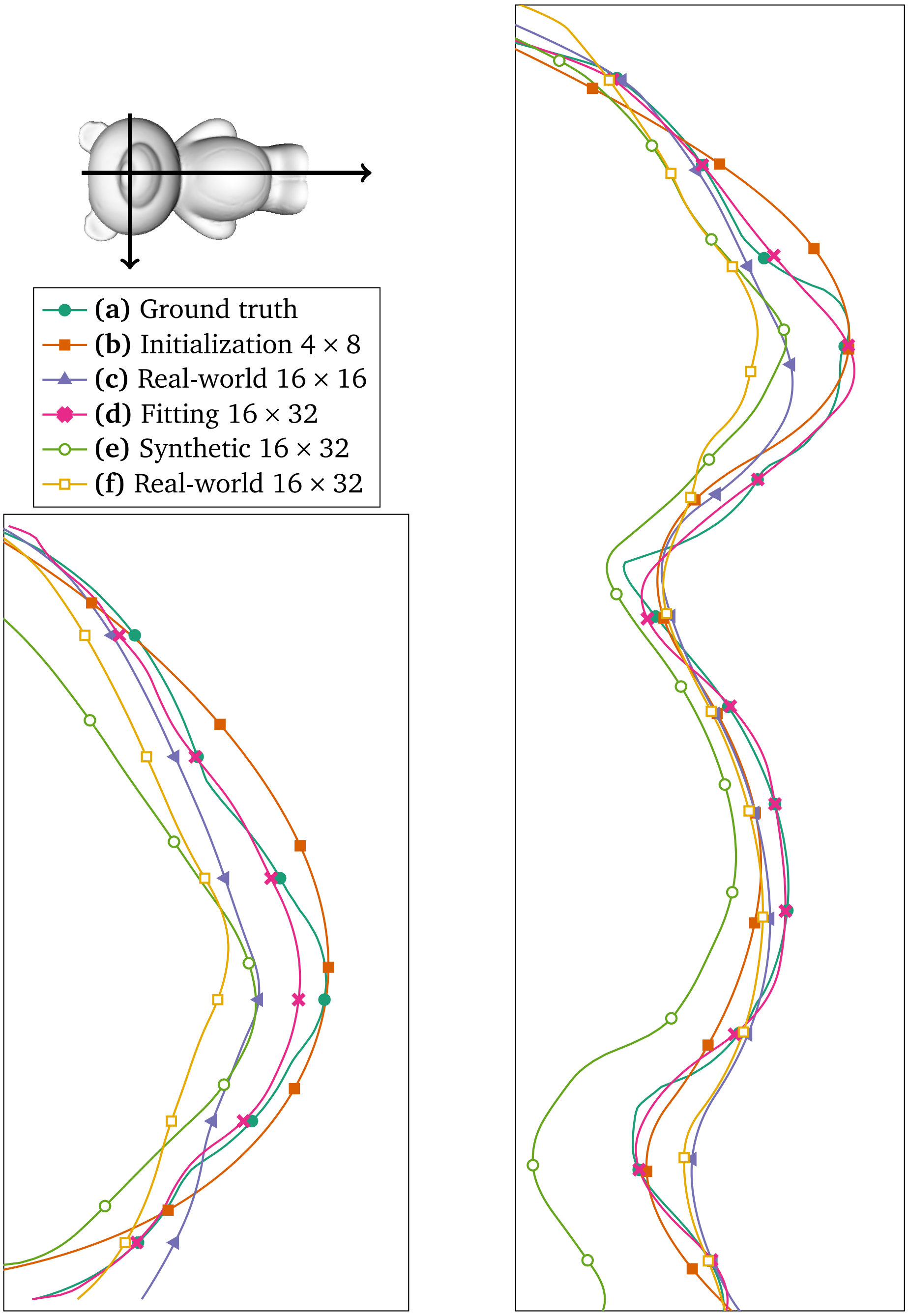
(c) Reconstruction (real-world input data, 16 × 16 parameters)

(d) Fitting to ground truth (16 × 32 parameters)

(e) Reconstruction of Synthetic-Teddy (16 × 32 parameters)

(f) Reconstruction (real-world input data, 16 × 32 parameters)

Figure 4.13: Rendering of the Teddy reconstruction results. Some surface parts are missing because the parts without any viewing rays were removed.



(a) Horizontal cut through the head and nose region.

(b) Vertical cut from top to bottom.

Figure 4.14: Horizontal and vertical cuts through the results of the TEDDY dataset. The plotted lines correspond to the renderings of Figure 4.13. Although the reconstructed surfaces drifted away from the initialization, the local shape of the ground truth geometry is visible.

Dataset name	Feature estimation	Surface parameters	Number of features	Iteration count	Runtime
CONVEXCONCAVE	known	7×7	12137	22	69 s
CONVEXCONCAVE	explicit	7×7	12137	21	75 s
CONVEXCONCAVE	implicit	7×7	12137	23	3.5 h
BUNNY	explicit	15×15	12824	22	30 m
MODELCAR	explicit	8×16	72	24	27 s
MODELCAR	implicit	8×16	72	59	7.6 m
TEDDY	explicit	16×16	31446	29	3.8 h
TEDDY	explicit	16×32	31446	21	15.8 h
SYNTHETICTEDDY	explicit	16×32	11875	19	4.3 h

Figure 4.15: Table of the runtime required for reconstruction of the presented datasets. Explicit feature estimation refers to Equation (3.10) whereas implicit feature estimation refers to Equation (3.11).

5 Discussion and Future Work

Recent work has shown that the reconstruction of specular surfaces under natural illumination is still a challenging problem. The methods evade the difficulties by setting up constraints to either the acquisition procedure or the kind of specular surfaces. In contrast, we have seen a novel framework in this thesis that is able to reconstruct specular surfaces based on sparse input data without the knowledge of feature locations in the environment.

5.1 Summary

The reconstruction is a minimization of an error based on the surface normals and the reflection vectors using a non-linear least squares optimization.

Synthetic experiments showed that it is possible to reconstruct a simple specular surface by modeling the reconstruction as an optimization problem using simple geometric residual functions. Accurate results are possible assuming known and unknown environmental feature positions. To transform the *inverse problem* with known feature positions into a problem with unknown environment, we either modeled these positions as optimization variables or used a different residual function which does not depend on the feature direction. We observed that the assumption of infinitely distant features can be violated without considerable large errors in the resulting mesh.

Real-world datasets were captured and an adapted image feature matching strategy was shown which is able to remove many outliers from the matching without the traditional epipolar geometry-based outlier tests. Using the matched reflectance correspondences, reconstructions containing local surface details similar to the ground truth could be achieved.

5.2 Limitations

The preparatory steps of the surface reconstruction, i.e. establishing feature correspondences, classifying features and initializing the surface geometry, are still a challenging task. Up to this point, no framework has been presented that is able to track or match image features of reflections on specular objects. One problem is the distortion of features which also leads to feature descriptors becoming dissimilar to each other. Even a simple tracking algorithm of corners seen in the reflection can be problematic because the corners change their shape leading to short feature tracks. Another problem is the robust filtering of outliers. Standard techniques, e.g. a ratio threshold on feature descriptor distances or RANSAC-based approaches trying to estimate a fundamental matrix, fail on image sequences of specular surfaces. The cause for this is that these techniques establish assumptions based on features being unique in one image or on epipolar geometry which are not given under these circumstances. Godard et al. [2015] have shown a way to circumvent the matching problem using an approach solely based on matching color information. They use a probabilistic technique to overcome the large amount of outliers emerging from the color value matching. In addition, they require a manual segmentation of the specular object and the background. However, the depicted matching problem still remains unsolved.

We have discovered that the proposed modeling of the reconstruction leads to an optimization problem with strong non-linearities. This property is due to the fact that the intersection point of the viewing rays with the surface is not known and fixed but varies during the optimization. As we have seen in synthetic experiments, this can be resolved using robust loss functions if the resulting reflection direction is different from the true direction. However, if the reflection direction coincidences with the true feature direction despite the wrong intersection point, problematic ambiguities arise which cannot be filtered out and hence influence the optimization.

Even if we can solve the matching problem—either manually or using carefully taken photographs with sharp reflections in combination with an adapted matching strategy—another problem emerged. We observe that in real-world experimental setups the natural environments do not necessarily provide features with an approximately uniform distribution. The proposed re-weighting of residuals lessens the effect but does not compensate for the poor feature distribution in all cases.

Nevertheless, it is possible to reconstruct specular surfaces if we start the iterative optimization using a good initialization. The initialization has to be accurate enough to lead the reconstruction problem to the correct local optimum. Godard et al. [2015] show that an initialization computed by voxel carving using the manually segmented specular object masks and the camera calibration is accurate enough to lead the optimization into the right direction. In this thesis, the initialization was computed using an MVS reconstruction of the edges and corners of the object. A smooth and low-dimensional Poisson surface extraction yielded an adequate initialization mesh.

In general, a high number of observations of a smaller set of environmental features is preferable instead of the reverse. The reason for this is the lower number of feature directions which have to be estimated during the optimization.

5.3 Future Work

As future work, a region-growing starting from confident reconstruction parts such as the MVS reconstruction of edges and corners should be tested. Using this approach, the confidence from already reconstructed parts could be transferred to nearby feature direction estimations. Another option would be to analyze other parametric surface models or an optimization based on a triangle mesh representation in combination with a regularization.

In the experiments, we have seen that the reconstruction can drift away from the initialization. To avoid this, the iterative reconstruction can be done in a coarse-to-fine way. The optimization could start with a surface consisting of a very low number of parameters. After convergence, the optimization is restarted using a surface with a higher number of parameters but with the same shape as in the previous iteration. In theory, this should robustify the optimization and steer the solution towards the global optimum.

Real-world objects often consist of different specular surface parts which are surrounded by diffuse structure or edges and corners, e.g. a car with its windows and painted body panels. An algorithm can segment these parts based on a traditional reconstruction and reconstruct each specular surface part using a local optimization. Furthermore, it would be possible to classify the specular objects and use prior knowledge to initialize the surface geometry.

Bibliography

[Agarwal et al.] S. Agarwal, K. Mierle, et al. Ceres Solver. URL http://ceres-solver.org. (p. 22)

[Agarwal et al., 2009] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a Day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72–79, Sept 2009. DOI: 10.1109/ICCV.2009.5459148. (p. 1)

[Balzer, 2008] J. Balzer. *Regularisierung des Deflektometrieproblems - Grundlagen und Anwendung*, volume 1 of *Schriftenreihe Automatische Sichtprüfung und Bildauswertung*. Universitätsverlag Karlsruhe, 2008. DOI: 10.5445/KSP/1000007844. (p. 9)

[Balzer and Werling, 2010] J. Balzer and S. Werling. Principles of Shape from Specular Reflection. *Measurement*, 43(10):1305–1317, 2010. ISSN 0263-2241. DOI: 10.1016/j.measurement.2010.07.013. (pp. 9 and 10)

[Balzer et al., 2011] J. Balzer, S. Höfer, and J. Beyerer. Multiview Specular Stereo Reconstruction of Large Mirror Surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2537–2544, June 2011. DOI: 10.1109/CVPR.2011.5995346. (p. 12)

[Blake, 1985] A. Blake. Specular Stereo. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'85, pages 973–976, San Francisco, CA, USA, 1985. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1623611.1623671. (p. 11)

[Blake and Brelstaff, 1988] A. Blake and G. Brelstaff. Geometry From Specularities. In *Computer Vision., Second International Conference on*, pages 394–403, Dec 1988. DOI: 10.1109/CCV.1988.590016. (p. 11)

[Blender Foundation] Blender Foundation. *Blender - a 3D modelling and rendering package*. Blender Institute, Amsterdam. URL http://www.blender.org. (p. 29)

[Fuhrmann et al., 2015] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele. MVE—An image-based reconstruction environment . *Computers & Graphics*, 53, Part A:44–53, 2015. ISSN 0097-8493. DOI: 10.1016/j.cag.2015.09.003. 40 years of Computer Graphics in Darmstadt . (pp. 2 and 18)

[Godard et al., 2015] C. Godard, P. Hedman, W. Li, and G. J. Brostow. Multi-view Reconstruction of Highly Specular Surfaces in Uncontrolled Environments. In *3DV*, 2015. DOI: 10.1109/3DV.2015.10. (pp. 3, 15, 36, and 48)

[Goesele et al., 2007] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz. Multi-View Stereo for Community Photo Collections. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. DOI: 10.1109/ICCV.2007.4408933. (p. 1)

[Jacquet et al., 2013] B. Jacquet, C. Häne, K. Köser, and M. Pollefeys. Real-World Normal Map Capture for Nearly Flat Reflective Surfaces. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 713–720, Dec 2013. DOI: 10.1109/ICCV.2013.94. (pp. 3 and 14)

[Kazhdan and Hoppe, 2013] M. Kazhdan and H. Hoppe. Screened Poisson Surface Reconstruction. *ACM Trans. Graph.*, 32(3):29–1, jul 2013. ISSN 0730-0301. DOI: 10.1145/2487228.2487237. (p. 37)

[Li et al., 2014] H. Li, T. Song, Z. Wu, J. Ma, and G. Ding. Reconstruction of a Complex Mirror Surface from a Single Image. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. McMahan, J. Jerald, H. Zhang, S. Drucker, C. Kambhamettu, M. El Choubassi, Z. Deng, and M. Carlson, editors, *Advances in Visual Computing*, volume 8887, chapter Lecture Notes in Computer Science, pages 402–412. Springer International Publishing, 2014. ISBN 978-3-319-14248-7. DOI: 10.1007/978-3-319-14249-4_38. (pp. 3 and 14)

[Liu et al., 2015] M. Liu, R. Hartley, and M. Salzmann. Mirror Surface Reconstruction from a Single Image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(4):760–773, April 2015. ISSN 0162-8828. DOI: 10.1109/TPAMI.2014.2353622. (p. 13)

[Lowe, 2004] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. DOI: 10.1023/B:VISI.0000029664.99615.94. (p. 18)

[Mann and Picard, 1994] S. Mann and R. W. Picard. Being ‘undigital’ with digital cameras: Extending Dynamic Range by Combining Differently Exposed Pictures. Technical Report 323, Boston, Massachusetts, 1994. Also appears, IS&T’s 48th annual conference, Cambridge, Massachusetts, May 1995. (p. 7)

[Nehab et al., 2008] D. Nehab, T. Weyrich, and S. Rusinkiewicz. Dense 3D Reconstruction from Specularity Consistency. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. DOI: 10.1109/CVPR.2008.4587683. (p. 12)

[Robertson et al., 1999] M. A. Robertson, S. Borman, and R. L. Stevenson. Dynamic range improvement through multiple exposures. In *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, volume 3, pages 159–163, 1999. (p. 7)

[Roth and Black, 2006] S. Roth and M. J. Black. Specular Flow and the Recovery of Surface Structure. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1869–1876, 2006. DOI: 10.1109/CVPR.2006.290. (p. 13)

[Sankaranarayanan et al., 2010] A. C. Sankaranarayanan, A. Veeraraghavan, O. Tuzel, and A. Agrawal. Specular Surface Reconstruction from Sparse Reflection Correspondences. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1245–1252, June 2010. DOI: 10.1109/CVPR.2010.5539826. (p. 13)

[Snavely et al., 2006] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring Photo Collections in 3D. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH ’06, pages 835–846, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6. DOI: 10.1145/1179352.1141964. (p. 1)

[Solem et al., 2004] J. E. Solem, H. Aanæs, and A. Heyden. A Variational Analysis of Shape from Specularities using Sparse Data. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 26–33, Sept 2004. DOI: 10.1109/TDPVT.2004.1335137. (p. 13)

[Swaminathan et al., 2002] R. Swaminathan, S. B. Kang, R. Szeliski, A. Criminisi, and S. K. Nayar. On the Motion and Appearance of Specularities in Image Sequences. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Computer Vision — ECCV 2002*, volume 2350, chapter Lecture Notes in Computer Science, pages 508–523. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-43745-1. DOI: 10.1007/3-540-47969-4_34. (p. 19)

[Tarini et al., 2005] M. Tarini, H. P. A. Lensch, M. Goesele, and H.-P. Seidel. 3D acquisition of mirroring objects using striped patterns . *Graphical Models* , 67(4):233–259, 2005. ISSN 1524-0703. DOI: http://dx.doi.org/10.1016/j.gmod.2004.11.002. (p. 9)

[Zisserman et al., 1989] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image and Vision Computing*, 7(1):38–42, 1989. ISSN 0262-8856. DOI: 10.1016/0262-8856(89)90018-8. (p. 12)

List of Figures