Advances in 3D Shape Acquisition

Diego Nehab

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
By the Department of
Computer Science

November 2007

Abstract

In this dissertation we discuss a variety of techniques that advance the state of the art in the field of 3D shape acquisition from real world objects. The research was done in collaboration with Szymon Rusinkiewicz, James Davis, Ravi Ramammorthi, and Tim Weyrich.

Our first contribution is a new framework for the classification of stereo triangulation algorithms. We classify methods according to the dimensions along which observations by both cameras are matched against each other. Different algorithms consider information that extends in space, in time, or simultaneously in both dimensions. Based on this framework, we design a novel algorithm for the triangulation of dynamic objects, as well as a new stereo setup based on unstructured active lighting.

We then present a novel sub-pixel precision refinement algorithm for stereo matches. We treat both cameras symmetrically, instead of assuming one camera to provide a reference image to be matched against. By refining match coordinates simultaneously on both cameras, we avoid a source of bias that can otherwise manifest itself as coherent noise in the reconstructions.

We also provide an efficient algorithm for combining position and orientation measurements into an optimal surface. Since position and orientation measurements are obtained from independent sources, each contains errors with distinct frequency characteristics. By optimizing a surface to conform to the most precise frequency components from each source, we can produce reconstructions that are substantially more precise than the original measurements.

Finally, we present a strategy for the acquisition of the 3D shape of shiny objects. Standard triangulation strategies that rely on captured appearances fail due to the view dependent nature of the images of such objects. We present a matching cost function based on surface normal consistency that can be used with standard dense stereo matching algorithms, and discuss the ambiguities that can arise.

Acknowledgments

It has been a privilege to work with Szymon Rusinkiewicz, my adviser. He was always there when I needed guidance, and never required me to be there unless it was necessary. I will look up to him for as long as I can discern him in the distance.

This dissertation would not have been possible without the help of James Davis, Ravi Ramamoorthi, and Tim Weyrich. In hindsight, five winters in Princeton were a small price to pay for a chance to collaborate with them. Thanks to Pedro V. Sander and Hugues Hoppe, who broadened my research interests to the point where our work did not fit within these pages, no matter how strong the stapler.

Thanks to the Princeton Graphics Group, past and present, for creating a great collaborative environment and an inspiring atmosphere. The comments I received from TIGGRAPH reviewers were always insightful and helpful.

I thank Princeton University, the Honda Research Institute, ATI Research, and Microsoft Research for funding my PhD. I also thank PUC-Rio for funding my undergraduate education, FAPERJ and CAPES for my MSc, and my parents for funding everything else.

Thanks to the Administrative Staff at the Department of Computer Science, especially Melissa Lawson, for making all rules and procedures seem straightforward. Thanks to the Computing Facilities Staff for keeping the systems going. Thanks also to Jennifer McNabb and Mladenka Tomasevic, from the Office of Visa Services, for helping me stay in the country. I also thank office 416 for having a window and a plant.

Thanks to all my friends here in Princeton. If I had thought I would need to write all your names down, I would have been less social.

Special thanks to Renato, for all the road paving, food testing, and reconnaissance. Thanks to Daniel and Davi for visiting me in Princeton and not alarming everyone back home about the dire conditions in which I lived. Thanks to Nir and Michael for sharing a roof with me. Juliana and Marcio (and now Luisa!) for providing me with a home away from home. Diogo and Leticia for giving me my godson, Gabriel. The gene pool needs more people like you.

Last, but not least, I would like to thank my family for their unrelenting support and encouragement. From now on it will be great to hear them say “Eu sempre soube que você ia conseguir!”, over and over again.

To my father.

Contents

List of Figures

Chapter 1

Introduction

A growing number of applications require computer models that represent the 3D shape of real world objects. A quality inspector, for example, may wish to verify that a certain part has been manufactured according to specification. A movie script may place actors in dangerous, unnatural, or otherwise impractical conditions, requiring a sequence to be simulated by computer. A surgeon may wish to replace a damaged part of a patient's bone with a prosthesis that exactly fits in its place. For many other examples, see section 1.1.

Although a skilled artist may be able to manually create these models, the budget, precision tolerance, and speed requirements of many applications can make this alternative unacceptable. Fortunately, a variety of methods have been developed that greatly reduce or completely eliminate the need for human intervention in the shape acquisition process. In fact, several companies (for example Cyberware, Minolta, Inspeck, and Eyetronics) currently market 3D scanning systems that produce complete models by directly measuring real world objects.

Despite the steady progress in the field, there is still no system or technique that produces satisfactory results under the entire range of conditions that might be important for different applications. In this dissertation, we present a variety of results that advance the state of the art in shape acquisition on different fronts. For example, capturing moving objects is in general more challenging than capturing static scenes. In chapter 2, we describe a technique that is suitable for this scenario. The methods described in chapters 3 and 4, on the other hand, focus on high-quality applications, such as the generation of photorealistic synthetic images. Shiny or glossy objects also present difficulties for traditional scanning methods, and we discuss an alternative for this case in chapter 5.

The goal of this introduction is to expose readers to the key techniques, mathematical concepts, and nomenclature that are required for a thorough understanding of the chapters that follow. Each chapter is self-contained, and we provide individual introductions that put them into context, as well as individual overviews of the related literature. Readers that are familiar with the area may choose to skip the introduction and move directly to the chapters of their interest.

1.1 Applications

In recent years, there has been a significant growth in the number of applications that take advantage of 3D models obtained from real world objects. This can be attributed to simultaneous advances in a variety of related fields. First, 3D scanners have become more affordable, more precise, and easier to operate. Additionally, the growing computational power of personal computers, especially due to advances in graphics hardware, has simplified the task of manipulating highly detailed models. Furthermore, advances in solid freeform fabrication technologies (for example, stereolithography, three-dimensional printing, and fusion deposition modeling) has made it straightforward for users to obtain synthetic solid objects directly from computer models. Below is a non-exhaustive list of some of the applications that benefit from these technologies.

Cultural heritage projects and archaeology: Perhaps the most conspicuous use of 3D scanning technologies has been in cultural heritage projects [17, 83, 69]. A variety of 3D scanners have been successful in producing models of famous relics and artifacts, notably of statues [77, 62, 48, 8, 54]. These models are often used by specialists for conservation and analysis purposes. Alternatively, within the context of museums [105, 122], the digitization of entire collections can be used to broaden public access via remote visualization techniques [47, 122]. Solid freeform fabrication can also be used to produce accurate replicas not only of individual artifacts, but of entire archaeological sites [1]. Computers can also assist in the reassembling of shattered items, including panels, pottery, and frescoes [33, 97, 71, 58].

Entertainment industry: Feature films, TV shows, and commercials are increasingly making use of acquired 3D models. The use of the technology is not clearly advertised by the movie industry, since its main use is in the production of convincing synthetic scenes that are nearly indistinguishable from reality. The scenes usually involve settings that would either be impractical to recreate in real life, would pose unnaceptable threats to the actors, or would be too expensive to produce. In such situations, actors can be scanned and animated, and the sequences can be generated by computer instead. Evidence of widespread use can be found in the portfolios of 3D scanner manufacturers (see for example Cyberware and Eyetronics [39, 46]). Computer games also frequently employ 3D models of human beings, particularly games based in movies or sports. In order to increase realism, such games often commission scans of an entire movie cast, or even of hundreds of athletes in a sporting league.

Other applications: In reverse engineering, CAD models can be quickly produced from 3D scans of machine parts. In computer aided inspection, manufactured parts are scanned to ensure they meet specifications. In robotics, 3D scanners are used to help in motion planning. In medical applications, dental prosthetics, bone replacement parts, and ear-canal hearing aids are made to perfectly match a patient that has been previously scanned. In architecture or civil engineering, as-built models are obtained with 3D scanners after the completion of construction work on buildings, industrial plants, bridges etc.

1.2 Shape acquisition

Producing a complete 3D model from an object is a multistage process. This process is traditionally divided into three stages: scanning, alignment, and merger. The scanning stage produces a partial model of the object, usually containing information about surface as visible to camera from a single viewpoint. Many scans of the object are captured, until every surface point (or the vast majority) is present in at least one scan (and hopefully in several). During the alignment stage, all scans are registered together, so that overlapping areas perfectly match. The multiple aligned scans are then finally merged into a single model.

Here we are mainly interested in the scanning stage, which is the original source of 3D shape information. In this section, we introduce two entirely different approaches to 3D scanning, which will be used in the following chapters. The first approach produces a dense collection position measurements that sample the surface of the object, and is introduced in section 1.2.2. The second approach measures the orientation of these surface points, and is introduced in section 1.2.3. Both acquisition strategies employ cameras to capture images of the desired object, and we therefore start by describing our camera model. At the end of the section, we briefly describe each of the remaining steps in the scanning pipeline. A longer discussion can be found, for example, in [6].

1.2.1 Camera model and calibration

The images captured by a camera are organized as matrices $I(x, y)$ of pixel intensities. Throughout this work, we assume the perspective projection camera model. This model relates the coordinates of a point $P = [X\ Y\ Z]^\top$, in a camera's reference frame, to the image coordinates $p = (x, y)$ of its projection according to the following formula:

$$(x, y) = (-f_x \frac{X}{Z} + c_x, -f_y \frac{Y}{Z} + c_y). \tag{1.1}$$

The parameters f_x and f_y, the horizontal and vertical focal lengths in pixels, and (c_x, c_y), the principal point, are collectively known as *intrinsic parameters*. These can be obtained automatically from any of a variety of widely available camera calibration toolkits [22, 92].

In practice, the calibration process also determines radial distortion coefficients, which account for pincushion or barrel distortion effects, as well as skew correction coefficients. Fortunately, these effects can be eliminated from the images by ressampling during a preprocessing stage, and therefore we do not have to include them in our camera model explicitly.

When more than one camera is used, which is often the case throughout this work, we also need the relative position between them. This information usually comes in the form of rotation matrices R_{ij} and translation vectors $\boldsymbol{t}_{ij}$, which transform the coordinates of a point P between the reference frames of each pair of cameras $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$:

$$P_j = R_{ij} P_i + \boldsymbol{t}_{ij}. \tag{1.2}$$

Collectively known as *camera extrinsics*, these parameters can also be obtained automatically from camera calibration toolkits. With this information at hand, we are ready to triangulate for positions.

1.2.2 Triangulating for positions

Consider the camera setup in figure 1.1. Given the image coordinates (x, y) of a surface point P as seen by a calibrated camera, we can define a ray through that point in the camera's reference frame as

$$P(Z) = \boldsymbol{p}\, Z = \begin{bmatrix} \frac{1}{-f_x}(x - c_x) \\ \frac{1}{-f_y}(y - c_y) \\ 1 \end{bmatrix} Z, \tag{1.3}$$

where Z, the depth of point P, is unknown.

If we have the image coordinates (x_i, y_i) of the same point P as seen by two different cameras, we can obtain two ray equations. These rays must meet at point P. Given the transformation between the two camera reference frames, as in equation 1.2, we can equate the two rays

$$R_{12}P_1(Z_1) + \boldsymbol{t}_{12} = P_2(Z_2) \tag{1.4}$$

to produce a linear system with three equations and two unknowns (Z_1 and Z_2), which we can solve by linear least squares to find the intersection. In practice, due to noise and other imprecisions, the two rays will not intersect. However, the closest point to either ray is usually an acceptable estimate for P.

Determining the corresponding image coordinates of a point P as seen by two cameras is significantly harder. Assuming that P projected to image coordinates (x_1, y_1) in camera $\boldsymbol{c}_1$, we must find the corresponding image point seen by camera c_2. Fortunately, due to the *epipolar constraint*, we do not have to consider all image points (x_2, y_2), but only those lying on an line.

Consider figure 1.1 again. From equation 1.3, we can determine the ray through pixel (x_1, y_1) in camera $\boldsymbol{c}_1$. Using equation 1.2, we can transform this ray equation to the reference frame of camera $\boldsymbol{c}_2$:

$$R_{12}\, \boldsymbol{p}_1 Z + \boldsymbol{t}_{12} = \begin{bmatrix} aZ + b \\ cZ + d \\ eZ + f \end{bmatrix}, \tag{1.5}$$

on which we introduced the constants a–f to avoid expanding the expression.

Now, using equation 1.1, we can project the ray into the image plane of camera $\boldsymbol{c}_2$. The resulting equations define the *epipolar line*, which we obtain by substituting the value of Z from the expression of x_2 into the expression for y_2:

$$(x_2, y_2) = \left(-f_{x_1} \frac{aZ + b}{eZ + f} + c_{x_1}, -f_{y_1} \frac{cZ + d}{eZ + f} + c_{y_1} \right) \Rightarrow a'x_2 + b'y_2 + c' = 0, \tag{1.6}$$

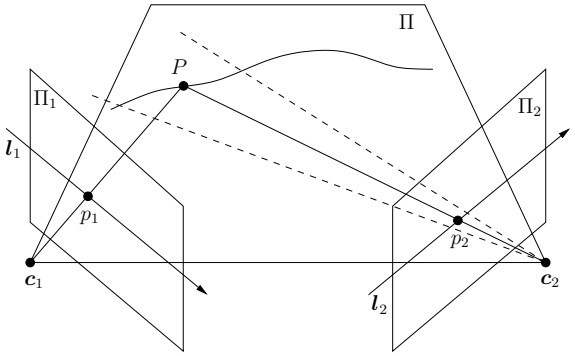


Figure 1.1: Triangulation and the epipolar constraint. A point P is observed by two cameras $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, projecting into the image planes Π_1 and Π_2 at image points p_1 and p_2, respectively. Given the image coordinates of p_1 and p_2, we can triangulate to find the 3D coordinates of point P. A point P, along with the centers of projection of cameras $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, defines a plane Π. Plane Π intersects the projection planes Π_1 and Π_2 to define lines $\boldsymbol{l}_1$ and $\boldsymbol{l}_2$. These are *epipolar lines*. From the construction, it is clear that if a point P projects anywhere in $\boldsymbol{l}_1$, it must project somewhere in $\boldsymbol{l}_2$. This is the *epipolar constraint*.

where again we introduced constants a', b', and c' to simplify out unnecessary details. Clearly, we can restrict our search for the correspondence of (x_1, y_1) to this line.

In practice, we use a convenient strategy known as *rectification* [80]. The idea is to warp the input images into the images that would be captured by two virtual cameras. The virtual cameras are parallel to each other and have matching focal lengths. By preserving the center of projection of the original cameras, the transformations can be expressed as homographies. Under the new configuration, however, the epipolar lines can be made horizontal, and can even share the same y coordinate on both images. This greatly simplifies the search for correspondences.

1.2.3 Normals from photometric stereo

The photometric stereo technique [126, 104] is one of the most straightforward methods for the acquisition of normal fields. In its simplest form, the method requires only three pictures of the object, lit by point light sources at known positions. Assuming a diffuse surface, the radiance measured by the camera at each pixel is proportional to $\boldsymbol{n} \cdot \boldsymbol{l}$, where $\boldsymbol{n}$ is the normal to the surface point being imaged, and $\boldsymbol{l}$ is a vector in the direction from the surface point towards the light source.

If we have pictures under three known light sources, we can setup one linear system per pixel, in the form

$$\begin{bmatrix} L_1\,\boldsymbol{l}_1 \\ L_2\,\boldsymbol{l}_2 \\ L_3\,\boldsymbol{l}_3 \end{bmatrix} \alpha\,\boldsymbol{n} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix}, \tag{1.7}$$

where the $\boldsymbol{l}_i$ are row vectors with the light source directions, the L_i account for the light source radiances, the E_i are the recorded image intensities at that pixel, and α is a measurement of surface

albedo that also accounts for the scaling of reflected radiance values as recorded by the camera. Both the $\boldsymbol{l}_i$ and the L_i can be obtained during calibration. We can therefore solve the system for $\alpha\,\boldsymbol{n}$, and obtain α by noting that $\boldsymbol{n}$ has unit length.

In practice, we position the light sources far away enough from the object, so that the light source direction and subintended solid angle do not vary significantly over the object surface. Finally, we use more than three light sources, and select among them the three best measurements, trying to avoid both shadows and specular highlights. The larger number of light sources also simplifies the task of ensuring the matrix in equation 1.7 is non-singular.

1.2.4 Surface representation

In computer aided design applications, it is common to represent surfaces as collections of piecewise polynomial patches. Each surface patch is defined by a set of control points in a grid, which can be edited by an artist. A continuous surface is obtained by the blending of polynomial basis functions, weighted by the control point coordinates, or alternatively as the limit of a subdivision process performed on the control grid. Medical applications, on the other hand, often deal with volumetric data, i.e., scalar fields defined over a volumetric domain. These data sets are usually acquired by Computed Axial Tomography (CAT scans) or by Magnetic Resonance Imaging (MRI scans). Such input data invite an implicit definition of shape. Surfaces are therefore defined as sets of domain points that form the preimage of a given measured value.

In the realm of 3D scanning, however, our data sources sample the object surface and produce a variety of measurements, such as 3D position, surface orientation, and color. We store these attributes directly in a list of vertices, and the surface can then be defined by connecting these vertices together to form a set of triangles. This vertex connectivity, which defines the set of triangles, is traditionally given explicitly or implicitly.

Range images: This strategy defines the triangles implicitly. Vertices are identified with the nodes of a two-dimensional regular grid. The grid usually follows the arrangement of the input images used to obtain the samples. Adjacent vertices are implicitly connected, and triangles are formed that tile the entire grid. In general, 3D scanners produce one range image per view of the object.

Triangle soups: This strategy defines triangles explicitly. Vertices are stored in an indexed list, and each triangle is given by a triad of vertex indices. This representation is more general and is usually adopted at later stages in the pipeline to define a single model that merges the information from many independent views.

1.2.5 Alignment and merger

In order to produce a complete 3D model of an object, several independent range images must first be aligned and then merged together into a single model. The alignment process can be divided into two stages: one that aligns pairs of overlapping scans and a global alignment stage.

The pairwise alignment is most often performed by one of many variations of the Iterated Closest Points (ICP) algorithm [10, 31]. The method starts with an initial hand alignment, and progressively refines it to find the rigid body transformation that minimizes an energy functional that measures the distance between the overlapping regions of the two surfaces being aligned.

When many range images are merged according to their pairwise alignments, the resulting errors tend to accumulate from one pair to the next as we move around the object. The goal of the global alignment stage is to evenly spread this error, reducing this accumulation effect [101]. More recent approaches consider non-rigid transformations during alignment, trying to account for the errors caused by miscalibration warps [27].

Once all range images have been aligned, we are left with a point cloud. Although such a representation is suitable for certain applications, the final product of a 3D scanning system is usually a triangle mesh that covers the entire surface of the object. The most popular methods for obtaining such a mesh are volumetric in nature [38, 70]. Combinatorial methods that attempt to interpolate a subset of the point cloud [7, 2, 72] tend to be less robust to noise.

1.3 Summary

Depth from triangulation has traditionally been investigated in a number of independent threads of research, with methods such as stereo, laser scanning, and coded structured light all considered separately. In chapter 2, which was written in collaboration with James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz [42], we propose a common framework called *spacetime stereo* that unifies and generalizes many of these previous methods. To show the practical utility of the framework, we develop two new algorithms for depth estimation: depth from unstructured illumination change and depth estimation in dynamic scenes. Based on our analysis, we show that methods derived from the spacetime stereo framework can be used to recover depth in situations where existing methods perform poorly.

Most dense stereo correspondence algorithms, such as the ones presented in chapter 2, start by establishing discrete pixel matches and later refine these matches to sub-pixel precision. Traditional sub-pixel refinement methods attempt to determine the precise location of points, in the secondary image, that correspond to discrete positions in the reference image. In chapter 3, which is based on a collaboration with Szymon Rusinkiewicz and James Davis [90], we show that this strategy can lead to a systematic bias associated with the violation of the general symmetry of matching cost functions. This bias produces random or coherent noise in the final reconstruction, but can be avoided by simultaneously refining both image coordinates in a symmetric way. We demonstrate that the symmetric sub-pixel refinement strategy results in more accurate correspondences by avoiding bias while preserving detail.

As outlined in section 1.2, we can obtain information about the geometry of surfaces in the form of either 3D positions (e.g., triangle meshes or range images) or orientations (normal maps or bump maps). In chapter 4, which is the result of a collaboration with Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi [91], we present an algorithm that combines these two kinds of estimates

to produce a new surface that approximates both. Our formulation is linear, allowing it to operate efficiently on complex meshes commonly used in graphics. It also treats high- and low-frequency components separately, allowing it to optimally combine outputs from data sources such as stereo triangulation and photometric stereo, which have different error-vs.-frequency characteristics. We demonstrate the ability of our technique to both recover high-frequency details and avoid low-frequency bias, producing surfaces that are more widely applicable than position or orientation data alone.

Finally, in chapter 5 we consider the dense reconstruction of glossy objects. The research is the result of a collaboration with Tim Weyrich and Szymon Rusinkiewicz. We propose the use of a specularity constraint, based on surface normal consistency, to define a matching cost function that can drive standard stereo reconstruction methods. We also present an aggregation method based on anisotropic diffusion that is particularly suitable for this matching cost function. Following a theoretical discussion on the types of ambiguity that can arise from the proposed constraint, we present a controlled illumination setup that includes a stereo camera pair, and one LCD monitor used as a calibrated, variable-position light source. We use the setup to evaluate the proposed method on real and synthetic data, and demonstrate its capacity to recover high-quality depth and orientation from specular objects.

Chapter 2

Space-time stereo triangulation

In this chapter, we consider methods that obtain depth via triangulation. Within this general family, a number of methods have been proposed including stereo [44, 110], laser stripe scanning [9, 37, 40, 65], and time- or color-coded structured light [5, 24, 55, 63, 111]. Although a deep relationship exists between these methods, as illustrated in the classification of figure 2.1, they have been developed primarily in independent threads of the academic literature, and are usually discussed as if they were separate techniques. We therefore present a general framework called *spacetime stereo* that helps in understanding and classifying different triangulation methods. By viewing each technique as an instance of a more general framework, solutions to some of the traditional limitations within each sub-space become apparent.

Most previous surveys classify triangulation techniques into *active* or *passive* methods [9, 36, 98, 118]. Active techniques, such as laser scanning and structured light, intentionally project illumination into the scene in order to construct easily identifiable features and simplify the task of establishing correspondences. In contrast, passive stereo algorithms attempt to find matching image features between a pair of general images about which nothing is known a priori. This classification has become so pervasive that we believe it is artificially constraining the range of techniques proposed by the research community.

We propose a different classification of algorithms for depth from triangulation that characterizes methods by the domain in which corresponding features are located. Techniques such as traditional laser scanning and passive stereo typically identify features purely in the *spatial domain*. In other words, correspondence is found by the analysis of the similarity of pixels in the image plane. Methods such as time-coded structured light and temporal laser scanning make use of features which lie predominantly in the *temporal domain*. That is, pixels with similar appearance variation over time are considered match. Most existing methods locate features entirely within either the spatial or temporal domains. However it is possible—and this paper will argue desirable—to locate features within both the space and time domains using the general framework of *spacetime stereo*.

The insight that triangulation methods can be unified into a single framework is the primary

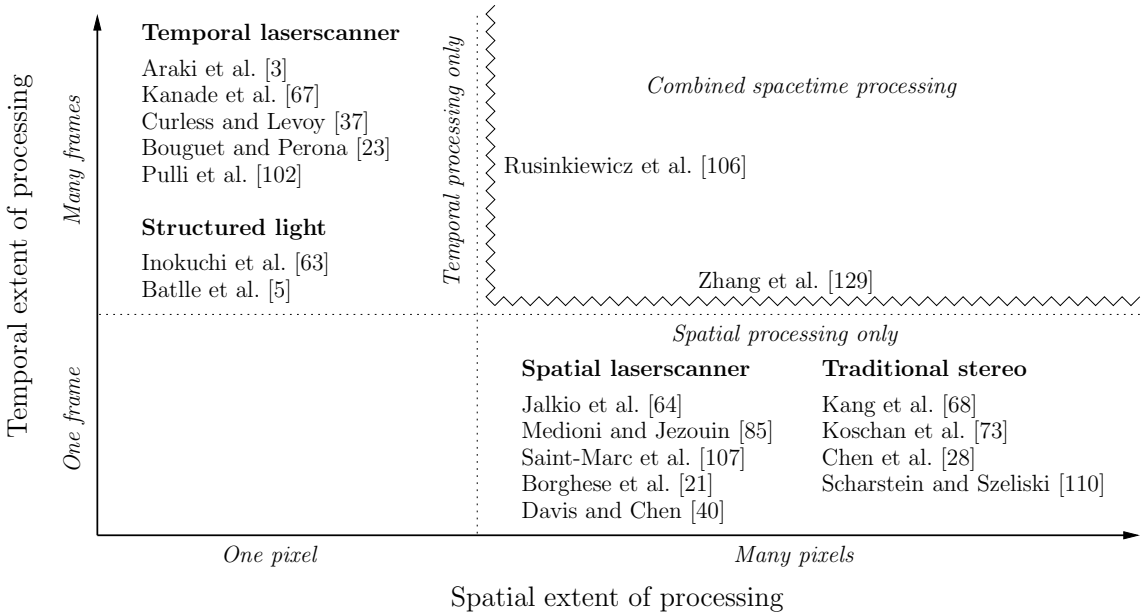


Figure 2.1: Most existing depth from triangulation techniques are specific instances of the more general spacetime stereo framework. Because these methods have been developed largely independently, they have often been artificially constrained to a small range of variation. Understanding that all these techniques lie in a continuum of possible methods can lead to previously unexplored combinations.

contribution of this chapter. The success of a proposed framework can be measured by its simplicity and its ability to bring new insights. We believe that this framework is sufficiently simple that most readers will find it intuitive and almost obvious in retrospect. To illustrate the framework's relevance, we show that it leads naturally to two new methods for recovering depth that have not been previously explored in the literature.

The first new method applies temporal processing to scenes in which the geometry is static, under uncontrolled illumination variation. We call this condition *unstructured light*, to distinguish it both from structured light methods in which lighting variation is strictly calibrated, and from passive stereo in which lighting variation is typically ignored. In our experiments, this variation is produced by the light and shadows from a handheld flashlight. The second new method applies spacetime processing to scenes in which the target object is allowed to move. In addition to evaluating the method, we analyze the necessity of spacetime processing, and show that optimal reconstruction is possible only by simultaneously using the spatial and temporal domains.

We are not alone in proposing that spatiotemporal information may be useful. Zhang et al. have simultaneously developed methods similar to ours, focusing on recovery of dynamic scenes rather than on constructing an organizing framework [130]. Other applications have been explored as well. For example, Shechtman et al. suggest that a spatiotemporal framework will be useful for increasing the resolution of video sequences [115].

This chapter is largely based on a journal article, in collaboration with James Davis, Szymon

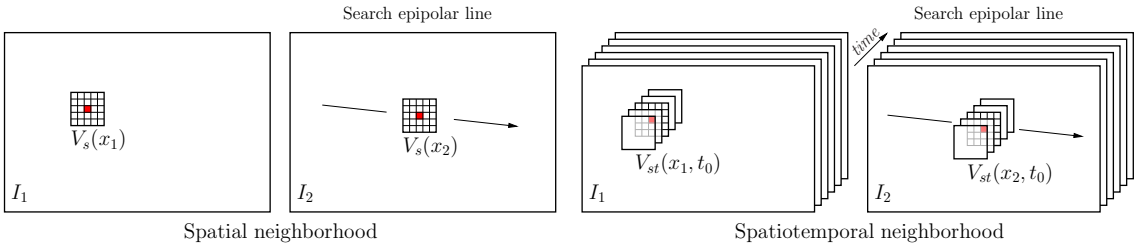


Figure 2.2: (Left) Traditional stereo matching. (Right) Spacetime stereo matching. In traditional stereo, the epipolar line is searched for similar spatial neighborhoods. In spacetime stereo, the matching windows include a temporal extent, and therefore the search is for similar spatiotemporal variation.

Rusinkiewicz and Ravi Ramamoorthi [42]. That article itself consists of a considerably expanded version of a previous conference paper [41, 42]. The journal article included new results on shape recovery for dynamic scenes, as well as a discussion of optimal spacetime windows in that context (as seen in section 2.5). The current text shows several improvements. In particular, we reran the static scene experiments of figure 2.5, in order to make the presentation consistent with the dynamic scene experiments shown in figure 2.8.

2.2 The spacetime stereo framework

The spacetime stereo framework can most naturally be understood as a generalization of traditional passive stereo methods. These methods proceed observing an object from two viewpoints in known positions, and attempting to find corresponding pixels in the two images. This search for correspondence can proceed either by comparing specific features, such as corners in each of the images, or more typically via matching of arbitrary spatial windows in the first image to corresponding regions along the epipolar line in the second image.

More specifically, traditional stereo finds correspondences by minimizing a matching function. In its simplest form, the cost can be written as

$$\left\| I_1(V_s(x_1)) - I_2(V_s(x_2)) \right\|^2. \tag{2.1}$$

Here I_1 is the intensity in image 1, I_2 is the intensity in image 2, and V_s is a vector of pixels in a *spatial* neighborhood around x_1 (or x_2). This is the standard minimization of sum of squared differences to find the best matching pixel x_2.

There is no reason to restrict the matching vector to lie entirely in a single spatial image plane. By considering multiple frames across time, we can extend the matching window into the temporal domain. The contrast between the two approaches is illustrated by figure 2.2. In general, the matching vector can be constructed from an arbitrary spatiotemporal region around the pixel in question. In the case of rectangular regions, a window of size $N \times M \times T$ can be chosen, where N and M are the spatial sizes of the window, and T is the dimension along the time axis. In this

general case, we seek to optimize the matching function

$$\left\| I_1(V_{st}(x_1, t_0)) - I_2(V_{st}(x_2, t_0)) \right\|^2. \tag{2.2}$$

Although the search for the best match follows the epipolar line, which lies entirely in the spatial domain, there is no mathematical distinction between the spatial and temporal axes, as far as the matching window dimensions are concerned. By choosing $T = 1$, we reduce the problem to traditional spatial-only stereo matching. By choosing $N = M = 1$ we use a purely temporal matching window. Under some conditions, a temporal matching vector is preferable to the traditional spatial vector, such as if the lighting in a static scene is changing over time. When there is motion, on the other hand, the spatial extent of the matching window becomes important. In general, the precise lighting and scene characteristics will determine the optimal size for the spacetime matching window.

2.3 Previous methods

Several well-investigated categories of research are in fact special cases of the general spacetime stereo framework discussed above. These include traditional stereo, time-coded structured light, and laser stripe scanning.

2.3.1 Traditional stereo

Traditional stereo matching is a well studied problem in computer vision. A number of good surveys exist [44, 110]. As discussed in section 2.2, traditional stereo employs matching windows that lie entirely in the spatial or image domain to determine correspondences.

Surprisingly, no existing stereo methods take advantage of the temporal domain. Presumably this is due to the ubiquitous classification of techniques into passive and active. Passive techniques are assumed to have no lighting variation, and thus no need for temporal processing. The framework and examples we present make clear that it is beneficial to extend existing stereo algorithms to use this additional source of information.

It should be noted that although epipolar analysis includes the language of "temporal" imaging, that work encodes camera motion on the temporal axis and is thus more closely related to multi-baseline stereo processing [18].

Most stereo methods can be broken into independent local matching and global regularization components. Since the ambiguities of passive stereo provide poor quality local correspondences, nearly all state of the art stereo research focuses on methods for global regularization, such as dynamic programming [93] or graph cuts [26]. In contrast, we focus on improving the local operator used for matching, using absolutely no method of global regularization. Many of the global methods for improved matching in the context of traditional spatial windows could be easily extended to include spatiotemporal windows for the local matching.

Methods for improving the local matching metric in stereo have also been proposed, such as adaptive windows [94] and robust matching metrics [13]. In this work we use very simple matching in order to isolate the importance of using the spacetime domain. In particular we use constant size rectangular windows, and accept the disparity that minimizes the SSD as shown in expression 2.2. More sophisticated matching metrics will of course improve the results beyond what we presented here.

Some stereo implementations make use of actively projected texture in order to aid the correspondence search. For example, Kang et al. use an uncalibrated sinusoidal pattern and reconstruct depth using a realtime multi-baseline solution [68]. We group techniques such as this with traditional stereo, rather than with the coded structured light methods discussed in the next section, because the correspondence search is inherently spatial, rather than temporal.

2.3.2 Time-coded structured light

Time-coded structured light methods determine depth by triangulating between projected light patterns and an observing camera viewpoint. A popular survey of these methods is by Batlle et al. [5]. The projector illuminates a static scene with a temporally varying pattern of light. The patterns are arranged such that every projected column of pixels can be uniquely identified. Thus, the depth at each camera pixel is uniquely determined based on the particular pattern observed at that pixel through time.

These systems rely on strictly controlled lighting, and most existing implementations are very careful to synchronize projectors, use one scanning system at a time, and remove ambient illumination. Our work unifies structured light methods with stereo matching, and thus eliminates the need for precise control over all aspects of scene lighting.

Although depth recovery in these systems is not typically described in terms of stereo matching, they do fall within the spacetime framework. The camera matching vector is purely temporal and is matched against a known database of projected patterns and their associated depths. The matching error metric can be written as

$$\left\| I_1(V_t(x_1, t_0)) - P_2(V_t(x_2, t_0)) \right\|^2, \tag{2.3}$$

which is similar to expression 2.2, except that we have replaced the second image I_2 with known projected patterns P_2. This is functionally equivalent to having a *virtual* second camera collocated with the projector. The virtual camera has the same viewpoint as the light source, so the virtual image it captures can be assumed identical to the projected light. By making conceptual use of a second camera, depth recovery in structured light systems can be described in terms of correspondence between images, similar to traditional stereo. It should be noted that the second camera need not be virtual. Using an additional real camera has a number of benefits, including improving the robustness of correspondence determination to variations in object reflectance [28], and generating high quality ground truth stereo test images [111].

2.3.3 Laser stripe scanning

Another alternative is laser scanning. A plane of laser light is generated from a single point of projection and is moved across the scene. At any given time, the camera can see the intersection of this plane with the object. Informative surveys have been provided by Besl [9] and Jarvis [65].

Most commercial laser scanners function in the spatial domain. The laser sheet has an assumed Gaussian cross section, and the location of this Gaussian feature is known in the laser frame of reference. Given a known laser position, the epipolar line in the camera image is searched for a matching Gaussian feature [107]. This match determines corresponding rays, and thus a depth value. Since the feature set lies only on one line in image space, rather than densely covering the image plane, only a single stripe of depth values is recovered. This process is repeated many times, while the laser stripe sweeps the entire object.

Laser scanners that function in the temporal domain have also been built [3, 67, 23]. As the laser sweeps past each pixel, the time at which the peak intensity is observed is recorded and used to establish correspondence. Curless and Levoy [37] provide an analysis of the benefits that temporal correlation provides over the traditional spatial approach in the context of laser scanning. Moreover, they show that the optimal matching uses feature vectors that are not strictly aligned with the time axis, but are "tilted" in spacetime.

As with coded structured light, laser scanning can be framed as standard stereo matching by replacing the calibrated laser optics with a second calibrated camera. With this modification, the laser stripe functions as the high frequency texture desirable for stereo matching. However, since the variation occurs in a small region, only a small amount (one stripe's worth) of valid data is returned at each frame. Two-camera implementations have been built that find correspondence in both the spatial [21, 40, 85] and temporal [102] domains.

2.3.4 Partial spacetime methods

As we have seen, most previous triangulation systems can be thought of as operating either in the purely-spatial or purely-temporal domains. Recently, however, researchers have begun to investigate structured light systems that make use of both space and time, though typically with many restrictions. One such system uses primarily temporal coding, adding a small spatial window to consider stripe *boundaries* (i.e., adjacent pairs of stripes) [55, 106]. Another approach uses a primarily spatial coding, adding a small temporal window to better locate stripes [129]. Still another approach considers "tilted" space-time windows that have extent in both space and time, but are only a single "pixel" thick [37].

Thus, as shown in figure 2.3, researchers have begun to explore the benefits of windows that are not purely spatial or temporal. However these methods were limited in the class of matching windows they considered, and expanding the domain of methods to encompass arbitrary space-time windows leads to improvements in robustness and flexibility.

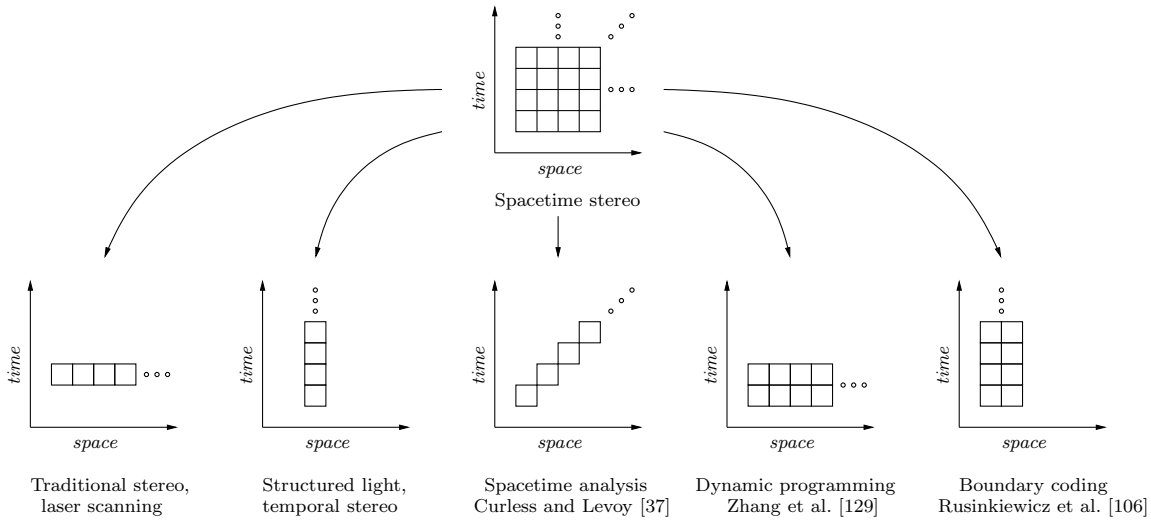


Figure 2.3: Previous triangulation methods can be rephrased as special cases within the spacetime stereo framework. Most methods use purely-spatial or purely-temporal matching windows, while a few others use hybrid, though restricted, window shapes.

2.4 Depth from unstructured illumination change

Consider the class of scenes which includes static objects illuminated by unstructured but variable lighting. This class includes scenes for which existing methods perform poorly, in particular textureless geometry lit by uncontrolled natural illumination, such as sunlight. Without resorting to global smoothness assumptions, traditional spatial stereo methods will not be able to recover any depth information in the textureless areas. On the other hand, active methods are not applicable because they rely on carefully controlled illumination patterns.

Spacetime stereo can recover high quality depth maps for this class of scenes. By analyzing reconstruction errors across the full range of possible spacetime window sizes, we can determine the best dimensions, which turn out to be purely temporal (i.e., *temporal stereo*). To illustrate the gains from this analysis, we present visual results showing that temporal stereo is capable of recovering depth with far greater accuracy than traditional spatial-only analysis. The reader should keep in mind that although temporal stereo is straightforward in light of the spacetime framework, it represents a truly new algorithm which has not been investigated previously.

2.4.1 Experimental setup

We used two scenes to evaluate our method, pictured in figure 2.4. One consists of blocks of wood, while the other contains a sculpture of a cat and a teapot. Stereo pairs were acquired using a single camcorder, with mirrors producing the two viewpoints. The working volume is approximately 50cm^3, and the viewpoints have a baseline separation of about 60°. Each viewpoint was manually calibrated using a target.

Figure 2.4: Sample stereo pairs for two of the scenes used in our experiments. The cat statue is illuminated by a flashlight, which was moving slowly over the scene. The blocks of wood were illuminated with a pattern of shadows cast by a hand in front of a fixed light source. Note the regions of uniform texture and lighting, which make traditional spatial stereo matching difficult.

We have experimented with a variety of different lighting configurations, moving a flashlight manually across the objects, moving a hand in front of a light source to cast a sequence of shadows, and using a hand-held laser pointer to illuminate the scene with a moving line. Under all these conditions, we were able to produce good reconstructions using spacetime stereo.

2.4.2 Spatiotemporal matching

In order to study the performance of spacetime stereo, we compared reconstructions of the data set of wooden blocks illuminated by a flashlight, using each possible spatiotemporal window size. Since ground truth is unavailable, we approximate it with the visually estimated best result obtained from the processing of a much larger data set of the same scene. For each spacetime window, we computed the fraction of wrong disparities, with regard to this "ground truth".

Figure 2.5 shows the results as a function of both spatial and temporal window sizes. For all spatial window sizes, we can see that increasing temporal window length is beneficial. There are no adverse effects from increasing the temporal length. In fact, the additional temporal information increases the probability of finding the correct match. Another insight, confirmed by the graph, is that after only a few frames of temporal information become available, it is no longer desirable to use any spatial extent at all: the lowest error was obtained using a spatial window of only a single pixel. This corresponds to the fact that spatial windows behave poorly near depth discontinuities.

It should be noted that the temporal order of frames in the video sequence was randomly shuffled to negate any effects caused by the specific path of flashlight motion. This has the effect of increasing the temporal information available in short temporal windows, since it removes correlation between neighboring frames. As a result, using a 1×1 spatial window becomes optimal after only 18 frames of temporal information are available. If we had not shuffled the frames, the number of frames required to outperform spatial stereo would have been higher, related to the

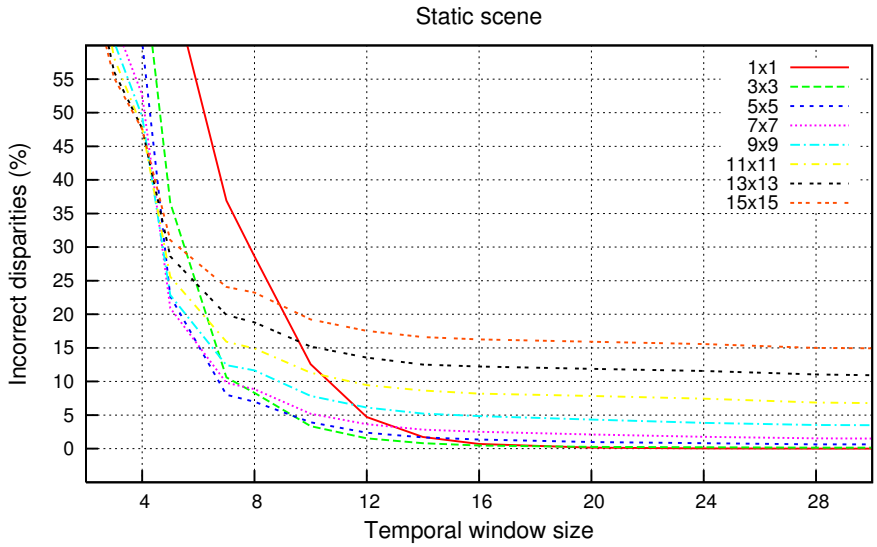


Figure 2.5: The the wood-block scene, illuminated with a flashlight, was reconstructed with a variety of matching window sizes. Results were compared against ground truth and the graph shows the percentual number of incorrect matches for each case. We conclude that for static scenes, longer temporal extents are always beneficial, and that best results can be obtained with a spatial window of 1×1.

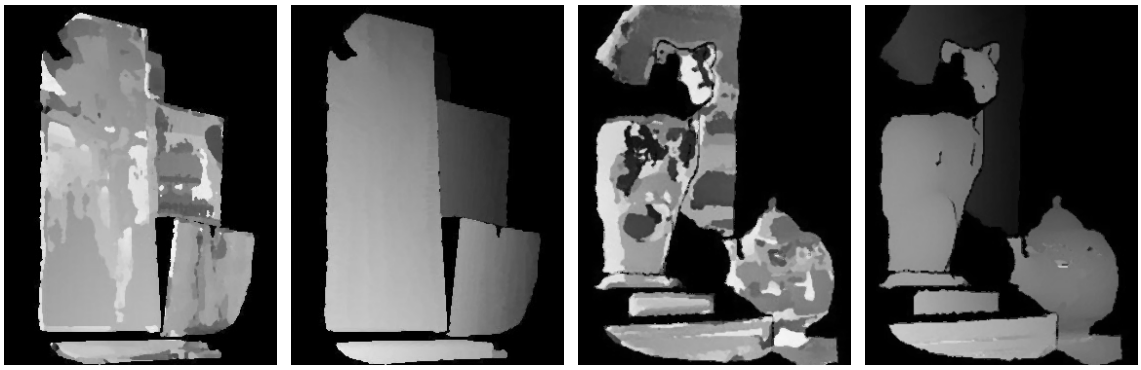
speed at which the flashlight moves. The original sequences in this case had approximately 64 frames, and 20 frames would have been enough to obtain a good approximation of depth.

Although an analysis of only one sequence is shown, we have recovered depth for hundreds of scenes and believe that the conclusions generalize. In particular, given static scene geometry and variable illumination, it is desirable to use a purely temporal matching vector.

2.4.3 Comparison of spatial and temporal matching

To show the practical utility of the spacetime stereo framework, we use our conclusions from the preceding analysis and compare purely spatial matching, as in standard stereo, with purely temporal matching. Spatial matching is computed using a 13×13 window; results were visually similar for other spatial window sizes. Temporal matching uses a single pixel, with a time neighborhood including the entire temporal sequence. Figure 2.6 shows the results.

We first consider the same sequence, in which wood blocks are illuminated with a flashlight. Spatial stereo matching is unreliable because the wooden blocks have large regions of almost uniform texture. Hence, the results are uneven and noisy. On the other hand, lighting variation creates texture in the time domain, making temporal matching robust. To show that our results generalize to a variety of conditions, we repeated the experiment using different geometry and lighting, a sculpted cat was subjected to shadowing. The results are similar: temporal matching produces much better results than spatial matching.



Spatial matching Temporal matching Spatial matching Temporal matching

Figure 2.6: Depth maps show a comparison between using spatial stereo matching with 13×13 neighborhoods against temporal stereo. (Left) The wooden blocks with lighting variation by manually moving a flashlight. (Right) The cat and teapot scene with lighting variation from shadows. Note that traditional spatial stereo depth estimates are uneven and noisy, while temporal stereo is relatively robust and accurate.

The scene of a white cat in front of a white wall was designed to be difficult or impossible for spatial stereo. Nevertheless some readers may be surprised that spatial stereo produces such poor depth estimates. We wish to reiterate that in order to compare only the proposed changes to local matching, no global regularization was used in these experiments. The addition of smoothness constraints would presumably improve the recovered depth regardless of whether spatial or temporal matching was used.

2.5 Depth of moving scenes

Scenes with motion represent a new set of challenges. Traditional passive stereo can process each frame of a sequence, but produces relatively low quality results. Active methods can not, in general be applied, since nearly all rely on a static object. Fortunately, the spacetime stereo framework is much more robust to motion. By subjecting the scene to high frequency illumination variations, a spacetime window can be used to recover depth. Although this is a straightforward application of the spacetime framework, it is unlikely that it would have been proposed by either the passive or active triangulation communities. The passive community would not propose active lighting, and the active community strictly controls lighting and does not speak in terms of stereo matching.

2.5.1 Experimental Setup

Moving objects require significantly higher-frequency (but still uncontrolled) lighting variation than static objects. In order to accommodate this need, we revised our experimental setup. A pair of cameras with a triangulation angle of approximately 15° were arranged to observe a working volume of approximately $30cm^3$. Instead of using a hand-held light source, an LCD projector was placed outside the camera baseline, but as nearby as is feasible, as shown on the left in

Figure 2.7: The experimental setup for moving scenes. (Left) Two synchronized cameras capture stereo views at 40Hz, while a projector displays random high frequency patterns at 60Hz. (Middle) A Sample stereo pair produced by the setup. (Right) The setup can capture static scenes with a quality comparable to a laser scanner, as the rendering shows.

figure 2.7. As before, the cameras were calibrated and synchronized with respect to one another, but the projector was left completely uncalibrated. Since the projected image can be changed at 60Hz, arbitrary high frequency lighting variation is possible. We simply projected random stripe patterns onto the scene. Our cameras are capable of capturing at approximately 40Hz. The middle of figure 2.7 shows a captured stereo pair. On the right is a reconstructed and rendered view of the object, captured while stationary. Note that although the lighting was unknown, the resulting accuracy is equivalent to a laser scanner.

In order to evaluate the optimal window size when objects are moving, we again needed ground truth data. Since this is not possible while an object is actually moving, we created "moving" data sets using stop-motion photography. The frog statue was moved by hand, under both linear and rotational motion, and a single image was taken at each position. When combined, these images simulate actual object motion. In order to obtain ground truth for a given frame, the frog was left stationary while additional lighting variation was projected and recorded.

2.5.2 Spatiotemporal matching

For each moving sequence, depth was computed using all possible combinations of spatiotemporal window sizes, and later compared to the ground truth. The percentage of correct wrong disparities was again obtained at each case.

For scenes with motion there is a trade-off in the temporal domain between obtaining additional information and introducing confounding distortions. If we repeat the analysis performed on static scenes, we expect U-shaped error curves, in which accuracy first improves and then decays as the temporal window size increases.

In the first condition, the frog was moved along a linear path at the rate of 1mm per frame. This is equivalent to roughly 3-4 pixels of motion in the image. Figure 2.8(top) shows the robustness of various window sizes. As expected, since the object is in motion, it is no longer preferable to use a very large temporal window. Disambiguating information must come from somewhere, and

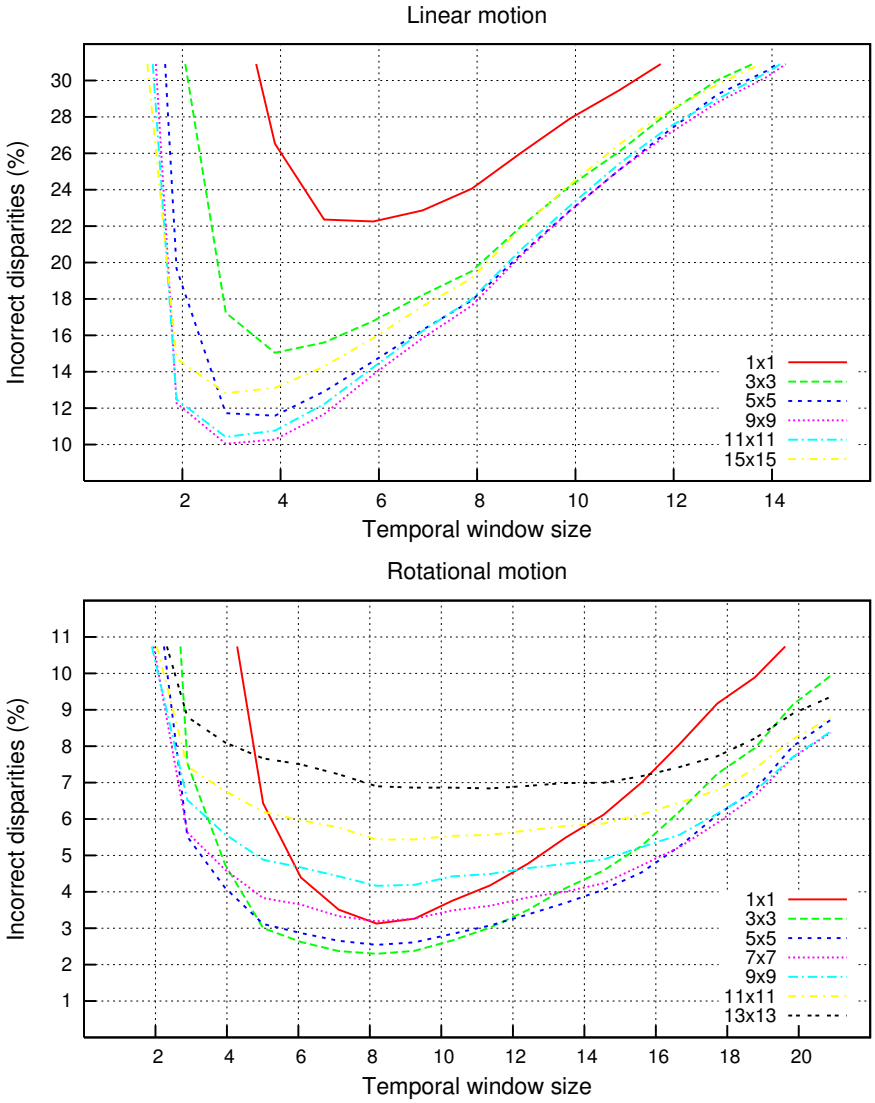


Figure 2.8: (Top) Matching error for a linearly moving scene as a function of temporal window size, for a variety of spatial window sizes. The result is a U-shaped curve for which the error first decreases with more disambiguating information, but then increases as motion makes matching difficult. Hence, a finite temporal window is desirable, and a $9 \times 9 \times 3$ spacetime window is seen to provide best results in this case. (Bottom) Matching error for a rotating scene, as a function of temporal window size for several spatial window sizes. The result is a U-shaped curve similar to the linear motion case. In this case, a $3 \times 3 \times 8$ spatiotemporal window is optimal, and is better than either spatial or temporal matching alone.

since the temporal window is smaller, a single-pixel spatial window no longer provides good results. In this case, we found a $9 \times 9 \times 3$ spatiotemporal window to be optimal. We also computed the optimum window size when the frog was subjected to rotation. When we used a rotation speed of 0.3° per frame, the optimal temporal window size was 8 frames long, and the spatial window size was 3×3. Figure 2.8(bottom) shows the same type of graph, but for the rotation scenario.

It is reasonable to wonder what would happen if the object moves either faster or slower. We increased the rotation speed by an order of magnitude to 3.0° per frame (a relatively high rate of rotation). Although the plot is not shown, the optimal temporal window size becomes very short, reducing to 2 frames. In this extreme case, object motion is so large that it is essentially best to treat each frame separately with spatial stereo.

The optimal window size is a function of the speed of object motion, the camera frame rate, the spatial texture available and the rate of temporal lighting variation. Although it is true that projected lighting will improve any stereo algorithm, we have shown that for some scenes *optimal* reconstruction requires the use of a spatiotemporal window.

When the object moves either very quickly or very slowly a degenerate form of spacetime stereo is optimal. For fast scenes spatial-only stereo is desirable, while for static scenes temporal-only stereo is desirable. Both spatial and temporal texture is desirable, and this texture should have a frequency roughly equivalent to the sampling frequency along that dimension.

2.5.3 Capturing motion

In order to demonstrate the capability of spacetime stereo on real dynamic scenes, we captured the motion of a deforming face. Rather than use stop motion photography as in the previous experiments, the cameras captured video at 40Hz, while the projector displayed stripe patterns at 60Hz. Depth was recovered at each frame of the sequence using a window size of $7 \times 1 \times 7$. This window size was chosen because both the horizontal and temporal dimensions have high frequency texture that is useful for matching. The vertical dimension (which is aligned with our stripe pattern) has relatively little texture, so does not contribute substantially to matching. This sequence cannot be reconstructed reliably using either spatial-only matching or temporal-only matching. The recovered depth was triangulated and several frames are shown rendered with lighting in figure 2.9a.

Rendered images of polygonal models with lighting are sensitive to the mesh surface normal. Since we show data prior to regularization it will appear noisy even if the error has low magnitude. Figure 2.9b visualizes the mesh after filling holes and smoothing the surface normals. These steps are analogous to the global regularization that is ubiquitously used in traditional stereo. Figure 2.9c shows a plot of the mesh depth along the line indicated above. Note that the noise level is well below 1mm.

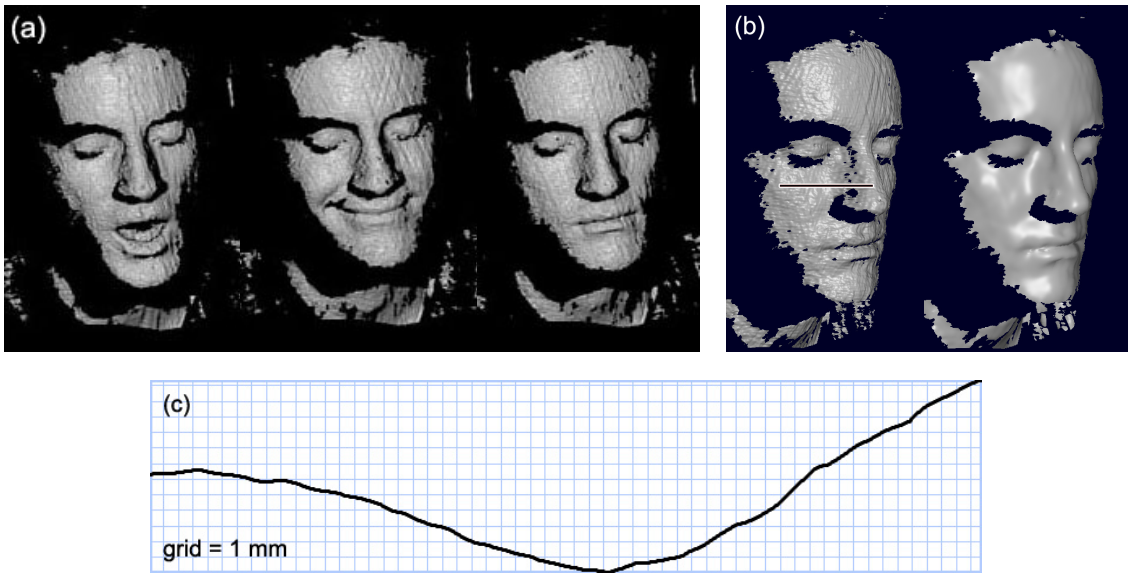


Figure 2.9: (a) Rendering of the 3D reconstructions of three frames in a dynamic scene that shows a face smiling (yours truly), captured at 40Hz. Note recovery of subtle features like the cheek deformation. (b) Recovered geometry before and after filling holes and smoothing mesh normals. (c) Plot of the mesh depth along the line indicated above. Note that although noisy normal estimates are perceptually distracting, the actual mesh geometry is accurate to under a millimeter, evident by visual inspection of the smoothness of this plot.

2.6 Conclusions

In this chapter, we presented a new classification framework, spacetime stereo, for depth from triangulation. Rather than distinguish algorithms as active or passive, we classify algorithms based on the spatial or temporal domain in which they locate corresponding features. This classification unifies a number of existing techniques, such as stereo, structured light, and laser scanning into a continuum of possible solutions, rather than segmenting them into disjoint methods.

As a demonstration of the utility of the spacetime stereo framework, we introduced two new methods: depth from unstructured illumination, and shape recovery for dynamic scenes using spacetime windows. In both cases, we demonstrated depth recovery results that are superior to those obtainable using traditional spatial-only stereo. Furthermore, we analyzed the optimal spacetime windows and showed that, for some classes of scenes, spacetime windows must be used for optimal reconstruction.

In summary, we believe that the spacetime stereo framework provides a useful way of thinking about many triangulation-based depth extraction methods, and the insights from it will lead to new applications.

Chapter 3

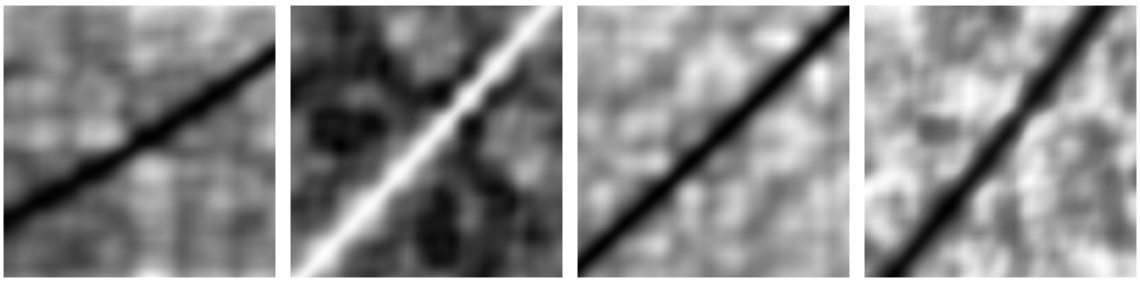
Symmetric Sub-pixel Refinement

The computation of precise sub-pixel stereo correspondences is vital to areas such as 3D scanning and image based modeling and rendering. Most dense stereo correspondence algorithms start by determining discrete pixel matches and later refine these matches to sub-pixel precision [110]. In global matching algorithms, the initial set of correspondences is usually computed by minimization of a *matching cost function* (such as the spacetime stereo metric described in chapter 2) that has been evaluated on an integer grid and stored in a *disparity space image* (DSI) [16, 128].

Sub-pixel refinement of correspondences can be performed over a finely sampled or continuously reconstructed neighborhood of the DSI around the initial integer match. The continuous reconstruction strategy has the advantage of being simple and efficient. On the other hand, although computationally more expensive, the supersampling alternative tends to be more accurate. Efforts have been made both to improve the quality of reconstruction-based refinement [116, 119] and to improve the efficiency of supersampling [49].

In this chapter, we identify a new source of bias for reconstruction-based sub-pixel refinement strategies (section 3.2). It can be observed when one image is considered as reference and the refinement is performed on the corresponding coordinate in the matching image. It arises from the sensitivity of this "traditional" approach to the varying confidence of the matching cost function when evaluated at neighboring pixels. In the final reconstruction, the bias can be experienced as random or coherent noise, as the "texture embossing" addressed by Curless and Levoy [37], or as the "striping effect" addressed by Zhang et al. [131].

To avoid bias, our symmetric sub-pixel refinement strategy refines both the reference and the matching image coordinates simultaneously, in a symmetric way, by looking for the minimum of the matching cost function along a direction that is insensitive to its confidence variations (section 3.3). We present results on both synthetic data and real scans, obtained using active stereo (section 3.4), which show that this new method significantly reduces bias in high-variance situations. Additionally, we demonstrate that one of its variants avoids the "pixel locking" effect addressed by Shimizu and Okutomi [116].



(a) (b) (c) (d)

Figure 3.1: Examples of matching cost functions. (a) Sum of squared differences. (b) Normalized cross-correlation. (c) Birchfield and Tomasi [12]. (d) Sum of absolute differences. Note the matching ridge and how the functions are symmetric with regard to it.

This chapter is largely based on a previously published conference paper [90], written in collaboration with Szymon Rusinkiewicz and James Davis. The current text brings a simplified proof for equation 3.6, as well as improved figures.

3.2 The symmetry of matching cost

Consider two rectified cameras C_1 and C_2, producing images I_1 and I_2 of an object, such that the scan-lines in each image are corresponding epipolar lines [80]. In this setup, Yang et al. [128] reduced the problem of stereo matching to that of finding a surface in the disparity-space image $\Xi(x_1, y, d)$, which measures the cost of matching points (x_1, y) in I_1 and $(x_1 + d, y)$ in I_2. The matching cost is defined by a metric M that compares neighborhoods of pixel values, so that $\Xi(x_1, y, d) \equiv M(I_1(x_1, y), I_2(x_1 + d, y))$. Note that for a given scan-line y, the problem simplifies even further to that of finding a *matching ridge*, which is the extremum curve in $\Xi_y(x_1, d)$.

Instead of working in disparity space, we prefer to work directly with image coordinates. The concept of disparity implies taking one camera as reference and, as we shall see, this is a source of bias. The direct parameterization $F_y(x_1, x_2) \equiv \Xi_y(x_1, x_2 - x_1)$ is more symmetric and simplifies our analysis. Figure 3.1 shows examples of popular matching cost functions under this direct parametrization. In each case, the matching ridge is clearly visible. We also notice a certain symmetry of the matching cost values, which we explain below.

Consider the intersection between the object being imaged and a given epipolar plane, as shown in figure 3.2. It defines a curve $O_y(t)$ that is projected into I_1 and I_2. If $r_1(t)$ and $r_2(t)$ are the corresponding parametrizations for these projections, the matching ridge is simply the curve defined by $R_y(t) = (r_1(t), r_2(t))$. Given a perfect matching pair (x_1, x_2), it is clear that R_y goes through (x_1, x_2) for some t. If r_1 and r_2 are continuous and smooth at t, then $(x_1 + \mathrm{d}r_1(t), x_2 + \mathrm{d}r_2(t))$ is a first order approximation for R_y. It follows that $(x_1 \pm \mathrm{d}r_1, x_2 \pm \mathrm{d}r_2)$ are also on the matching ridge and therefore are also matching pairs.

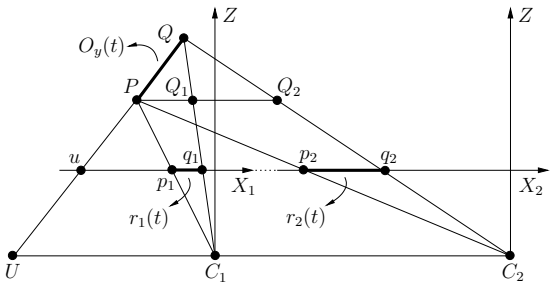


Figure 3.2: The slope of the matching ridge. The geometry of the setup yields an expression for the slope $\mathrm{d}r_2/\mathrm{d}r_1$, as given by equation 3.6. Each intersection U between the object tangent and the baseline of the cameras produces a different slope.

Comparing the values of $F_y(x_1 + \mathrm{d}r_1, x_2 - \mathrm{d}r_2)$ and $F_y(x_1 - \mathrm{d}r_1, x_2 + \mathrm{d}r_2)$, we notice that they must be similar:

$$\begin{aligned}
& F_y(x_1 + \mathrm{d}r_1, x_2 - \mathrm{d}r_2) \equiv \\
& \quad \equiv \quad M(I_1(x_1 + \mathrm{d}r_1, y), I_2(x_2 - \mathrm{d}r_2, y)) && (3.1)\\
& \quad = \quad M(I_2(x_2 - \mathrm{d}r_2, y), I_1(x_1 + \mathrm{d}r_1, y)) && (3.2)\\
& \quad \approx \quad M(I_1(x_1 - \mathrm{d}r_1, y), I_1(x_1 + \mathrm{d}r_1, y)) && (3.3)\\
& \quad \approx \quad M(I_1(x_1 - \mathrm{d}r_1, y), I_2(x_2 + \mathrm{d}r_2, y)) && (3.4)\\
& \quad \equiv \quad F_y(x_1 - \mathrm{d}r_1, x_2 + \mathrm{d}r_2) && (3.5)
\end{aligned}$$

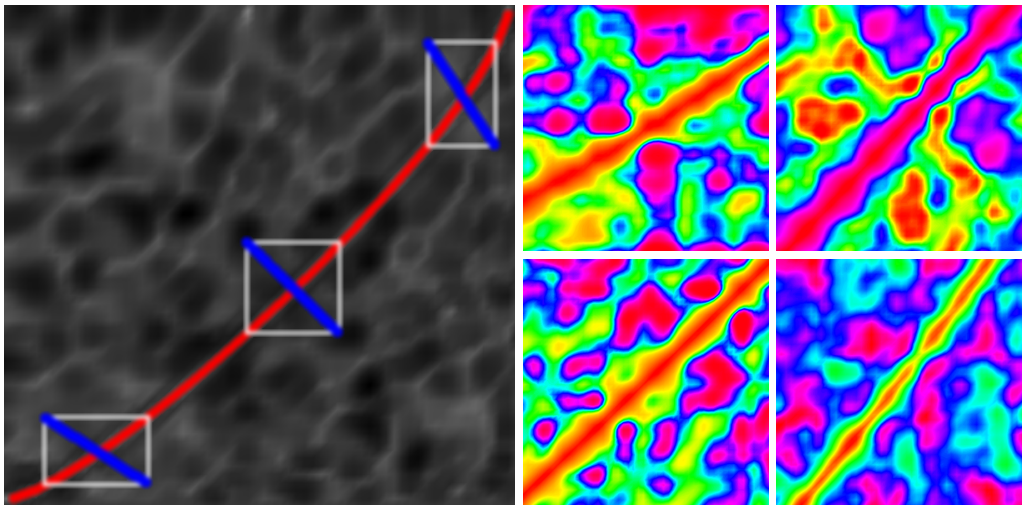
Steps 3.1 and 3.5 are by definition. Step 3.2 follows from the symmetry of M. Steps 3.3 and 3.4 come from the fact that, since $x_1 \pm \mathrm{d}r_1$ matches $x_2 \pm \mathrm{d}r_2$, $I_1(x_1 \pm \mathrm{d}r_1, y)$ must be similar to $I_2(x_2 \pm \mathrm{d}r_2, y)$. The continuity of M then leads to the approximations.

We have thus shown that F_y is locally skew-symmetric about R_y. The symmetry is such that, if a segment of the matching ridge is the diagonal of a rectangle, then symmetric pairs can be found along *symmetric lines* parallel to the other diagonal. These may or may not be perpendicular to the matching ridge (see figure 3.3).

The slope of the matching ridge (which is also the symmetry axis) is given by $\frac{\mathrm{d}r2}{\mathrm{d}r1}$. This ratio can be written as a function of the baseline distance between the cameras and the tangent to the object at the point being imaged (equation 3.6). For a geometric derivation, assume a linear object segment PQ as in figure 3.2. The slope $\frac{\mathrm{d}r_2}{\mathrm{d}r_1}$ is equal to the ratio between the lengths of segments $\frac{p_2q_2}{p_1q_1}$. From triangles PC_1Q_1 and PC_2Q_2, we conclude that $\frac{p_2q_2}{p_1q_1} = \frac{PQ_2}{PQ_1}$. By the same reasoning on triangles UQC_1 and UQC_2, we reach $\frac{PQ_2}{PQ_1} = \frac{UC_2}{UC_1}$. We have thus proved that

$$\frac{\mathrm{d}r2}{\mathrm{d}r1} = \frac{UC_2}{UC_1} = \frac{Z - X_2\frac{\mathrm{d}Z}{\mathrm{d}X_2}}{Z - X_1\frac{\mathrm{d}Z}{\mathrm{d}X_1}} \tag{3.6}$$

The slope is $\frac{\pi}{4}$ if the two cameras coincide or if the object tangent is parallel to the baseline. When the tangent goes through either center of projection, the ridge is perpendicular to the



Symmetry axis Symmetric structures

Figure 3.3: The skew-symmetry of matching cost. (Left) If a segment of the matching ridge (shown in red) is the diagonal of a rectangle (shown in white), symmetric pairs can be found along the other diagonal (shown in blue). (Right) The symmetry becomes clear by applying a colormap to the sample matching cost functions of figure 3.1. Even structures considerably far away from the matching ridge tend to have correponding features on the other side.

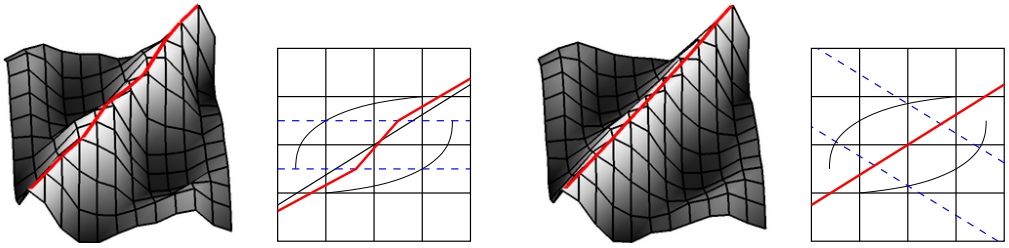
corresponding axis. Interestingly, any object tangent going through a given point U in the baseline produces the same matching ridge slope. When U falls between C_1 and C_2, only one of the cameras can see the object, because the other camera sees it from behind. It follows that the slope of the matching ridge is always positive.

With these observations in mind, we proceed to an analysis of the traditional approach to sub-pixel refinement.

3.2.1 Traditional sub-pixel refinement

Assume (i_1, i_2) is a pair of integer best matches. Considering C_1 as the reference camera, the traditional approach is to determine a sub-pixel precision correspondence $i_2 + \bar{t}_2$ for each i_1. To determine the optimal value of the displacement $\bar{t}_2$, a continuous constant-i_1 cut is reconstructed from the matching cost function values neighboring (i_1, i_2), and $\bar{t}_2$ is chosen to bring the reconstruction to its extremum. It is common practice to fit a parabola or other curve to the values $F_y(i_1, i_2 - 1)$, $F_y(i_1, i_2)$ and $F_y(i_1, i_2 + 1)$.

Unfortunately, a closer look at the matching ridges of figure 3.3 reveals that their "widths" vary considerably. This variation reflects the changing confidence of matching cost functions when applied to different pairs of epipolar points. For example, uneven surface texture and illumination might cause the matching function to be highly discriminating in one part of the surface, leading to a higher and narrower ridge. Elsewhere, the matches might be more ambiguous, leading to a lower and flatter ridge.



(a) (b) (c) (d)

Figure 3.4: Uncertainty bumps. (a) Sliding an axis-aligned cut across uncertainty bumps causes bias. (c) On the other hand, cuts aligned to the symmetric lines of the matching cost function are insensitive to the bumps. Figures (b, d) show schematic views of the real data shown in figures (a, c). Curved lines show a hypothetical uncertainty bump and dashed lines show the cut directions.

As we compute the sub-pixel matches for each i_1, we slide the constant-i_1 cut past several of these *uncertainty bumps* in the matching ridge. As each bump goes by, the fit is biased first to one side, then to the other. Figure 3.4a shows the phenomenon in real data, and figure 3.4b explains why it happens schematically. This bias is responsible for most of the noise seen in the "traditional" reconstructions of figures 3.6 and 3.7.

As suggested by figures 3.4c and 3.4d, we can avoid this problem if we look for the extrema along the symmetric lines of the matching cost function. Neither camera is considered as reference, and the refined matches will have sub-pixel precision in the coordinates of *both* images. This is the fundamental idea behind our symmetric sub-pixel refinement method.

3.3 Symmetric sub-pixel refinement

Guided by the desire to capture the symmetry of the matching cost function, we consider a 2D neighborhood of matching cost values around (i_1, i_2), and reconstruct a continuous *surface* $\mathcal{S}(t_1, t_2)$ from it. We then define $\mathcal{C}(t) = \mathcal{S}(s_1 t, s_2 t)$, a cut through the reconstruction in the $[s_1 \;\; s_2]^\top$ direction. The symmetric sub-pixel refined match is given by the pair $(i_1 + s_1\bar{t}, i_2 + s_2\bar{t})$, where $[s_1 \;\; s_2]^\top$ follows the lines of symmetry of matching cost, and $\bar{t}$ is chosen to bring the cut to its extremum.

All that is left to do is choose the surface reconstruction method and find the direction of the cut. Below we investigate some options.

3.3.1 Quadric interpolation

One candidate for reconstruction is a quadric that interpolates all 9 values in the 3×3 neighborhood N_3 around (i_1, i_2). This quadric is uniquely defined by the following formulas:

$$\mathcal{S}_q(t_1, t_2) = \mathbf{q}^\top(t_2)\, N_3\, \mathbf{q}(t_1), \tag{3.7}$$

$$\mathbf{q}(t) = \begin{bmatrix} \frac{1}{2}t(t+1) \\ 1 - t^2 \\ \frac{1}{2}t(t-1) \end{bmatrix}. \tag{3.8}$$

Note that, under this reconstruction, the traditional approach of fitting a parabola to the constant-i_1 cut reduces to finding the extremum in the $[0\ 1]^\top$ direction.

Since a cut through a quadric is at most a degree 4 polynomial, there is a closed form expression for the position of its extremum. However, since the initial bracket is trivial and the target precision is modest, the extremum can be easily and efficiently determined with a golden section search [99].

3.3.2 Uniform B-spline approximation

Moving away from interpolation, we can consider a larger neighborhood and use a B-Spline approximation for the matching cost function. Consider a 5×5 neighborhood around (i_1, i_2). It is composed of four overlapping 4×4 neighborhoods N_4^j. We can define cubic patches for each of these, and use their union as the B-Spline approximation:

$$\begin{aligned} \mathcal{S}_b(t_1, t_2) &= \mathcal{S}_b^j(t_1 - o_1^j, t_2 - o_2^j), && (3.9)\\ \mathcal{S}_b^j(t_1, t_2) &= \mathbf{b}^\top(t_2)\, N_4^j\, \mathbf{b}(t_1), && (3.10)\\ \mathbf{b}(t) &= \frac{1}{6}\begin{bmatrix} t^3 \\ -3t^3 + 3t^2 + 3t^2 + 1 \\ 3t^3 - 6t^2 + 4 \\ -t^3 + 3t^2 - 3t + 1 \end{bmatrix}. && (3.11) \end{aligned}$$

The offsets o_1^j and o_2^j simply adjust (t_1, t_2) to local patch coordinates. Note that S_b is C^2 continuous everywhere, and only the 3×3 neighborhood around (i_1, i_2) influences the surface at the center of the parametrization.

3.3.3 Gaussian cylinder approximation

Since the matching cost function neighborhoods we are interested in are part of the matching ridge, we can design a surface with meaningful degrees of freedom. To this end, the following surface represents a Gaussian Cylinder generated by a straight line:

$$\begin{aligned} \mathcal{S}_g(t_1, t_2) &= G(D(t_1, t_2)) && (3.12)\\ G(d) &= a\mathrm{e}^{-d^2} + b, && (3.13)\\ D(t_1, t_2) &= s_1 t_1 + s_2\, t_2 - p. && (3.14) \end{aligned}$$

This surface enforces a ridge-like shape for the reconstruction. The parameters a, b, s_1, s_2, and p can be determined by non-linear least squares minimization on the 3×3 neighborhood around (i_1, i_2). The line $D(t_1, t_2) = 0$ then gives the local approximation for the matching ridge, from which the sub-pixel estimate can be easily found. Usually, a few iterations of the Levenberg-Marquardt method, as implemented by Lourakis [81], are enough for a good fit.

3.3.4 Choice of cut direction

As suggested by figure 3.4, the direction $[s_1\ s_2]^\top$ that follows the symmetric lines of the matching cost function is the right choice for a cut through $\mathcal{S}$. Besides respecting the symmetry of matching cost, this direction will in general be more stable than axis aligned directions. Unfortunately, since formula 3.6 requires previous knowledge about the scene, we can not directly use it to determine the cut direction.

We notice, however, that the principal direction of highest curvature of $\mathcal{S}$ at $(0, 0)$ provides a good estimate for $[s_2\ s_1]^\top$. This is because the highest curvature happens for cuts almost perpendicular to the matching ridge. From that, $[-s_1\ s_2]^\top$ is an approximation for the matching ridge direction and $[s_1\ s_2]^\top$ is therefore a good estimate for the direction we are looking for.

In practice, this is how we obtain the cut direction for the quadric interpolation and the B-Spline approximation. For the Gaussian cylinder, the estimate is not required (it is directly available from the surface parametrization).

3.4 Results

To evaluate our method, we tested it with real and synthetic data, using a temporal stereo triangulation scanner setup [42, 130]. In this active scanning technique, random stripe patterns are projected onto the scene while two cameras capture synchronized images. Since each point in the visible surface receives a unique light profile through time, it is possible to establish correspondences in a fashion similar to the area-based matching of standard stereo, but using windows that extend only through time (i.e., with spatial width and height of just one pixel). This strategy has the advantage of producing perfect correspondences and of being unaffected by depth discontinuities. It provides us with a way to isolate the sub-pixel refinement evaluation from other sources of error that could mask the effects we want to analyze.

Our real scanner is composed of two Sony DFW-X700 1024×768 firewire cameras and a Toshiba TLP511 projector with the same resolution. The cameras are calibrated using the toolbox by Bouguet [22] and synchronized by an external trigger. Our virtual scanner uses similar camera parameters, but produces image pairs from a 3D model, simulating the stripe patterns with projective textures. Both scanners have an estimated resolution of 0.2mm and a working volume 2000 times as large. Our tests are performed with static scenes, using sequences of 32 images, and with the normalized-cross-correlation metric. Fewer images would be sufficient, but the additional information improves the quality of the matching cost function. Figure 3.5 shows examples of input images to our system. The close-ups shown correspond to two of the pairs used to produce the object reconstruction on the left of figure 3.6.

We use two synthetic reference models: a sphere, for its wide range of smooth depth and orientation variation, and a parametric surface $Z(r, \theta)$, for its sharp features and arbitrarily small details:

$$Z(r, \theta) = -\frac{1}{10} r |\sin 16\,\theta|. \tag{3.15}$$

Figures 3.6 and 3.7 show renderings of data recovered by the virtual and real scanners respectively, using the traditional parabolic fit, the method by Shimizu and Okutomi [116], and our symmetric method employing each of the proposed reconstruction alternatives. The figures show that our method eliminates most of the visible noise equally well for each reconstruction alternative. In particular, note how the "striping effect" has been eliminated from the Greek panel in figure 3.6.

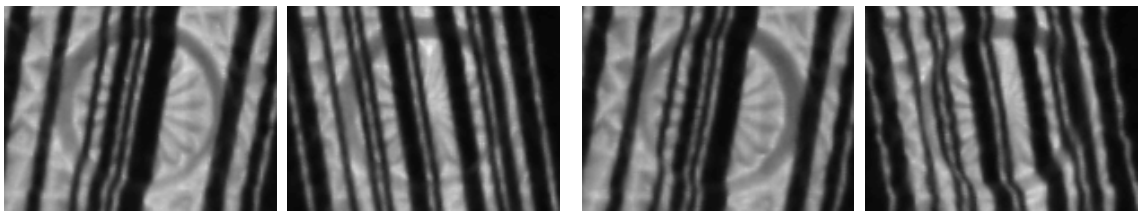
The Gaussian cylinder reconstruction also reduces the "pixel locking" effect addressed by Shimizu and Okutomi [116]. This is no surprise, since a similar result holds for the traditional sub-pixel refinement with Gaussian fit [103]. Using the spherical model, we computed histograms of the estimated sub-pixel displacement from the integer match. Results are shown in figure 3.8.

Figures 3.9 and 3.10 show depth profiles for the reconstructed synthetic sphere and parametric surface. The profile for the 5mm radius sphere shows a considerable variation in object tangent direction (about 115 degrees). This in turn produces large variations in the matching ridge slope. The spherical profiles show that our method performs well across such variations. In the parametric surface, sharp details are progressively smaller closer to the center. The profiles show that our method recovers details up to the same resolution as the traditional approach. Therefore, it is not simply eliminating noise at the expense of detail.

Figure 3.11 shows the "texture embossing" effect on the depth profile of a real planar object whose reflectance varies sinusoidally. The varying reflectance causes the confidence of the matching cost function to vary wildly along the matching ridge. Accordingly, severe uncertainty bump errors disrupt the traditional sub-pixel refinement strategy. In contrast, the noise levels observed in the symmetric reconstructions are within the expected scanner precision.

3.5 Conclusions

In this chapter we identified a new source of bias in the sub-pixel refinement of stereo correspondences. In reconstructed scenes, the bias manifests itself as random or coherent noise. To avoid this bias, we presented a novel technique that exploits the inherent symmetry of matching cost functions and refines matching coordinates in both images simultaneously. Results show that our approach performs better than previous techniques.



Left camera Right camera

Figure 3.5: Examples of stereo input images. A close-up is shown from two of the image pairs used in the reconstruction of the object shown on the left of image 3.6. Patterns of varying orientation are required to ensure that no ambiguities arise when the projector is placed away from the baseline of the cameras.

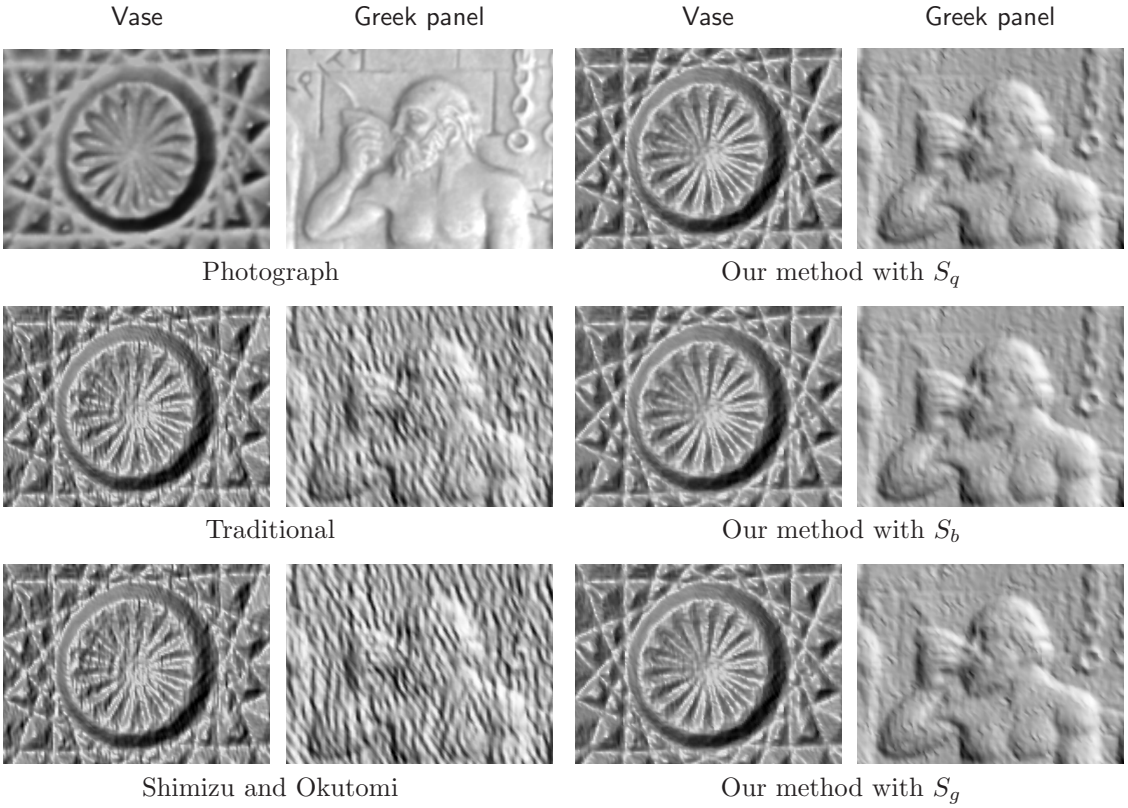


Figure 3.6: Renderings from reconstructed geometry for the real scanner. Note how the noise level is reduced by our method. In addition, note how the "striping effect" was eliminated from the (replica) Greek panel scan.

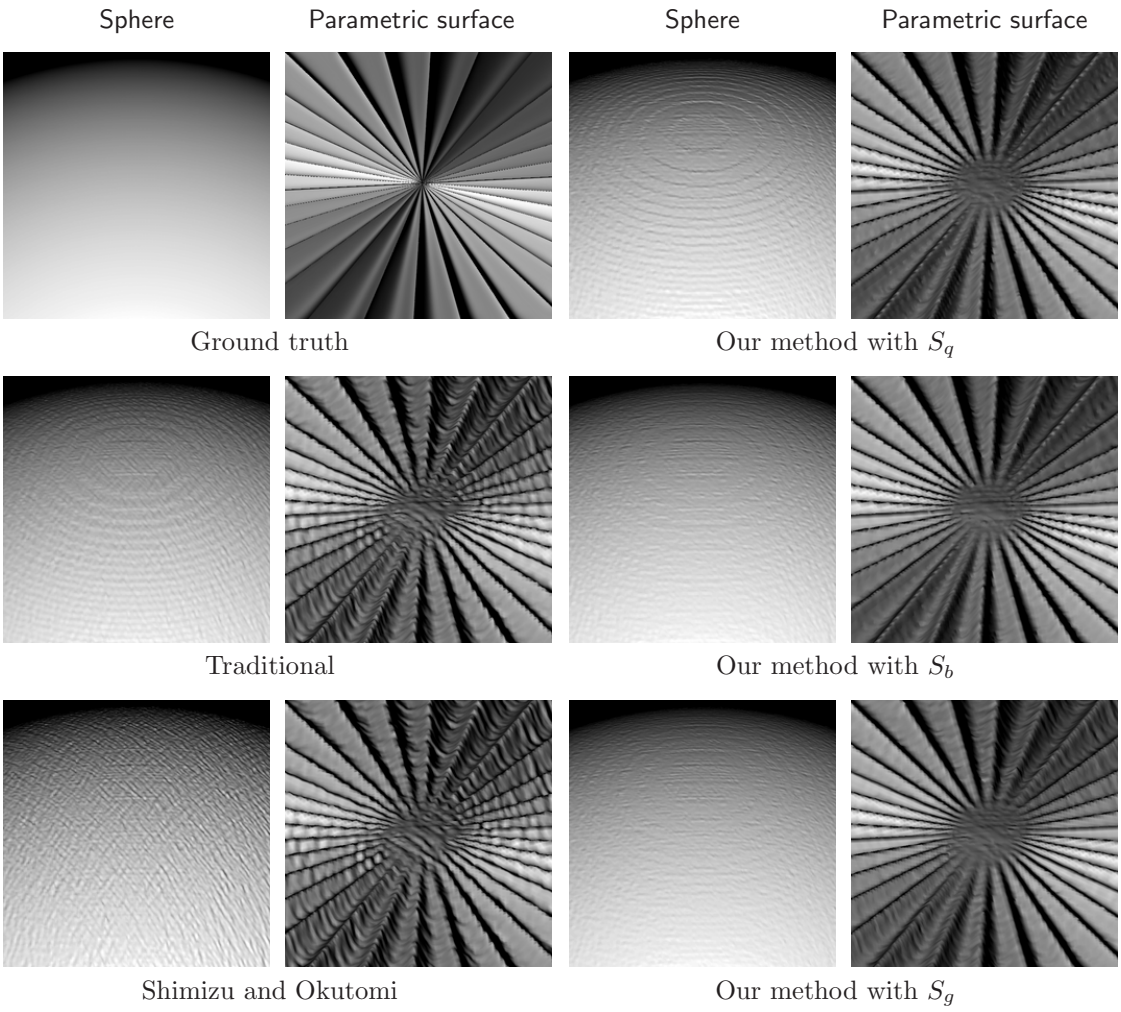


Figure 3.7: Renderings from reconstructed geometry for the virtual scanner. From the spherical model renderings, note how S_g reduces the "pixel locking" effect. From the parametric surface, note how detail is preserved while noise is eliminated.

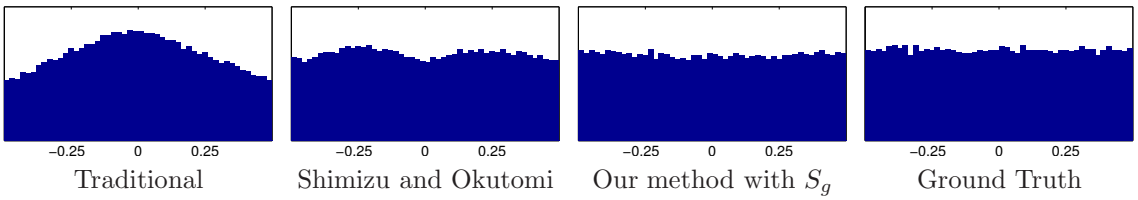


Figure 3.8: Histograms of sub-pixel deviation from the integer match for the spherical model. The traditional method is biased towards the center, producing a "pixel locking" effect. The method by Shimizu and Okutomi [116] performs better, but is still biased towards ± 0.25. In contrast, the histogram for the Gaussian cylinder reconstruction is almost flat, as is the ground truth.

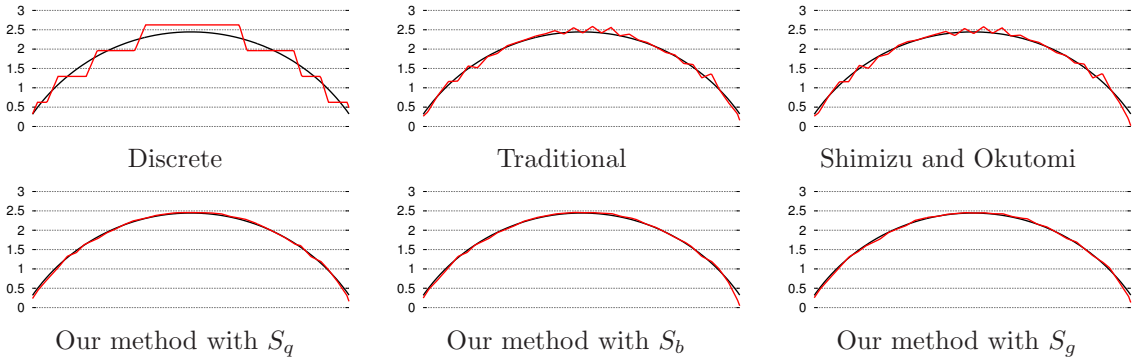


Figure 3.9: Depth profiles for a synthetic spherical model (5mm radius). For each plot, ground truth is shown in black. Note the reduced noise level for a variety of matching ridge slopes (produced by the varying object tangent).

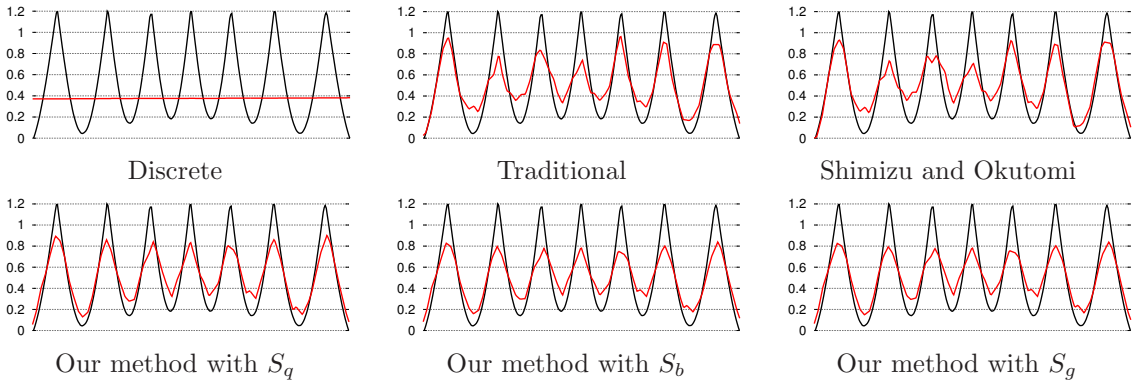


Figure 3.10: Depth profiles for the synthetic parametric surface. For each plot, ground truth is shown in black. The profiles show that our method does not simply eliminate detail along with noise.

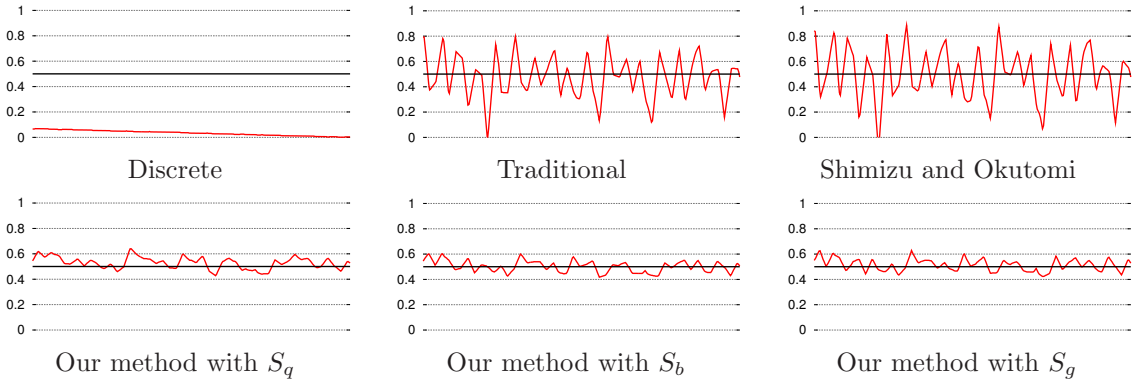


Figure 3.11: Depth profiles for a *real* planar object whose reflectance varies sinusoidally. The least-squares fit plane is shown in black. The varying albedo generates severe systematic biases in the traditional sub-pixel estimation. On the other hand, the noise observed in the symmetric reconstructions is within the scanner precision.

Chapter 4

Combining Normals and Positions

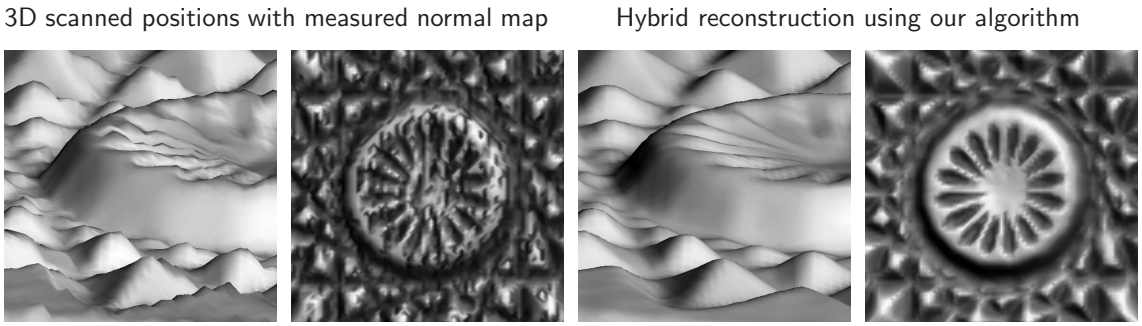
As we discussed in the introduction, scanned 3D models of real-world objects are being used in rendering, visualization, and analysis applications. Although the absolute accuracy of such scanned data can be satisfactory, even a small amount of noise in measured positions can cause large errors when surface normals are computed. Therefore, lit renderings of such scanned models may produce low quality images, as seen in figure 4.1a.

Instead of computing normals from the measured positions, we can directly measure an independent normal field. As we have shown in section 1.2.3, there exist technologies based on *shape from shading* or *photometric stereo* [126] that directly measure surface orientations. Using these independently-measured fields as normal maps allows for high-quality renderings from certain viewpoints (see figure 4.1b), even when the actual mesh has low quality or low resolution. One drawback to simply pasting a "good" normal field onto "bad" geometry, however, is incorrect parallax and occlusion at grazing views (figure 4.2a). In addition, some rendering and mesh processing effects such as shadowing or accessibility shading [86] inherently operate on only the surface positions, not the normals. Thus, the poor performance of such techniques on noisy geometry (as shown in figure 4.2b) cannot be directly ameliorated by the availability of high-quality normals. Although it is sometimes possible to directly integrate the high-resolution normal field to produce a surface without using additional geometric information, the lack of constraints between multiple disconnected patches, as well as the frequent presence of low-frequency distortion (as shown later in the chapter), can lead to bias in the reconstruction.

In this chapter, we present a hybrid algorithm that produces a surface that optimally conforms to given surface positions and normals, taking advantage of both sources of information. Figures 4.1 and 4.2 show results of running our algorithm. Although most of our examples focus on combining the depth information from a triangulation scanner with the normal information from photometric stereo, the technique is applicable to positions and normals (or bump maps) acquired through scanning, manual editing, or signal processing algorithms. Our method is efficient on dense, real-world datasets, since we formulate our optimization in a way that requires solving only a sparse linear system. In addition, only the most reliable frequency components of each source

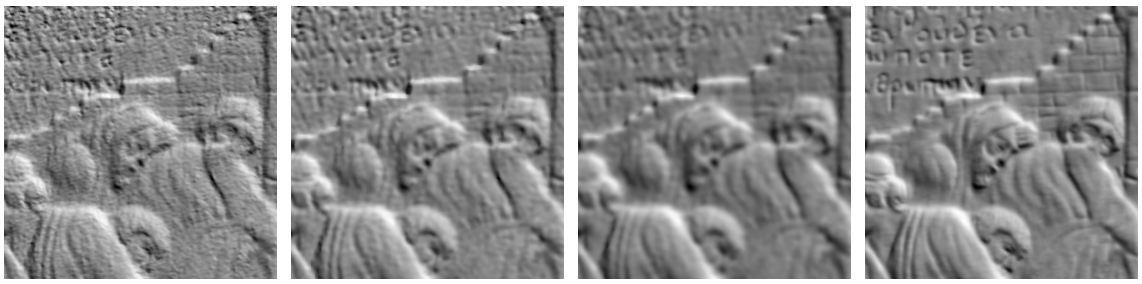
(a) (b) (c) (d)

Figure 4.1: (a) Rendering of 3D scanned range image, (b) same scanned geometry, augmented with a measured normal-map (from photometric stereo), (c) our hybrid surface reconstruction, which combines both position and normal constraints, (d) photograph. Note how our method eliminates noise from the range image while introducing real detail. The surface normals computed from the hybrid geometry are of the same quality or better than those from photometric stereo, while most of the low-frequency bias has been eliminated.



(a) (b) (c) (d)

Figure 4.2: Rendering at grazing angles (a) and accessibility shading (b). High-quality normal maps are not appropriate in these cases. The examples require the precise geometry our method can produce with the same input data (c, d).



(a) (b) (c) (d)

Figure 4.3: Range scanners frequently have noise in position measurements that is on the order of sample spacing, leading to noise in estimated normals (a). The popular approach of downsampling or smoothing the raw measurements either leaves remaining noise (b) or blurs out relevant surface detail (c). In contrast, our method combines measured positions with measured normals, leading to noise elimination while adding real detail (d).

are considered, resulting in a reconstruction that both preserves high-frequency details and avoids low-frequency bias.

Our method is motivated by an analysis of the common error characteristics of measured normals and positions (section 4.3). It proceeds in two stages: first, we correct for low-frequency bias in the measured normal field with the help of measured surface positions (section 4.4.1). Then, we optimize for the final surface positions (sections 4.4.2–4.4.4) using linear constraints and an efficient sparse solver. We analyze the performance and quality of the algorithm on a variety of datasets (section 4.5), demonstrating its ability to reconstruct accurate and precise geometry suitable for rendering and mesh processing.

This chapter is based on a previous conference paper, written in collaboration with Szymon Rusinkiweicz, James Davis, and Ravi Ramamoorthi [91]. The most important addition to the current text is the improved formulation for the full model optimization, presented in section 4.4.4, which replaces the version presented in the original paper. We also include new results in figures 4.8 and 4.11. Finally, the new implementation of the algorithm is freely available for download by interested researchers.

4.2 Relation to previous work

Positional measurement: Of the many range scanning technologies available (see [9, 98, 35] for surveys), methods based on triangulation have become popular, since they can be flexible, inexpensive, and accurate. However, a fundamental limitation of triangulation-based methods is that their depth accuracy is some fixed multiple of their (horizontal) sample spacing, which typically results in noisy estimated normals. Although downsampling or blurring the raw measurements is sometimes acceptable, it frequently leads to oversmoothing of detail (figure 4.3). The only effective means of reducing noise is to combine multiple measurements. Curless and Levoy [38] have investigated a volumetric method for combining multiple range scans, showing that their VRIP algorithm is essentially a least-squares estimator that can average away noise while keeping detail. In contrast, we investigate combining the fundamentally different data types of measured positions and measured normals. Although Kazhdan et al. [70] have recently proposed a Poisson formulation for the volumetric reconstruction problem using a set of points with normals as input, their work focuses on obtaining a watertight surface from a point cloud, rather than on obtaining a precise surface estimate from uncorrelated position and normal measurements.

Orientation measurement: The photometric stereo method [126], which obtains surface orientation from shading information, is part of a larger set of methods known as *shape-from-shading* [57]. These include methods whose outputs are surface normals, or sometimes simply single components of surface normals. Although it is possible to integrate the normals to find the shape of the surface, this approach is fragile when accurate surface reconstruction is desired, largely because integrating normals is prone to introducing low-frequency biases (recall that the action of the integration operator in frequency space is to scale energy at some frequency ω by an

amount proportional to $^1/_\omega$, hence exaggerating any low-frequency noise). Furthermore, when the surface consists of multiple disconnected components, integration has no way of determining the relative position of those components. Therefore, most of the use of measured surface orientations in graphics has been in rendering, by directly using the measured normals as normal maps [8], which produces effective results in some but not all applications (figure 4.2).

Combining positions and orientations: Much of the existing work on combining measured positions and orientations is intended for rough surface reconstruction in computer vision [123, 4, 50, 75], rather than for accurate reconstruction of surfaces for rendering or mesh manipulation. Accordingly, orientation measurements are taken into account mostly because dense estimates are available, and can help fill holes left by stereo correspondences based on sparse features.

Of methods that are general, one class integrates normals to yield a surface, then merges the resulting mesh with the measured positions as a final step. As mentioned above, this typically has the side effect of introducing bias and robustness problems. Some of these methods [34, 87] address the problem of low-frequency deformation by performing frequency-dependent processing. Although our method has a similar effect, it is inherently more stable in the presence of bias and disconnected components, as are other techniques that *avoid* explicit integration of normals.

A final class of methods combines positions and normals by formulating the reconstruction as a nonlinear optimization of constraints provided by the different measurements [59, 60, 29]. However, for the typical scanned meshes used in computer graphics, which contain 10^5 polygons and above, nonlinear optimization methods can be prohibitively expensive. Our formulation of the optimization using a sparse, typically diagonally-dominant, linear system results in efficient optimization for large meshes, such as the ones shown throughout this chapter.

4.3 Motivation and quality assessment

In this section we motivate our method by showing some of the typical error characteristics of acquired positions from depth scanners and acquired normals from photometric stereo techniques. It should be noted that, although we focus on this application, our method can be applied to a wider range of scanning, modeling or mesh processing techniques, being independent of the specific methodology or experimental setup. In the remainder of this section, we first describe the specific scanner we used in this work, followed by an assessment of the quality of recovered positions and normals.

4.3.1 Experimental setup and hybrid scanner design

Consider the setup shown in figure 4.4. To acquire positions, we use a temporal stereo triangulation scanner (such as in chapter 2), which combines the accuracy of active triangulation with the ease of calibration of stereo. It is composed of two Sony DFW-X700 firewire cameras and a Toshiba TLP511 projector. The cameras are calibrated using the toolbox by Bouguet [22], and synchronized

by an external trigger. The projector flashes a series of random stripe patterns onto the object while the cameras simultaneously capture images from two different viewpoints.

Because each surface point on a given epipolar plane receives a unique lighting profile through time, it is possible to establish unique correspondences by simply correlating the intensity variations in both images. There is no need to calibrate or even synchronize the projector with the cameras. Given the correspondences, the 3D positions of each point as seen by either camera can be determined by triangulation.

To acquire normals, we augment our scanner with a number of fixed light sources and use the photometric stereo technique [126]. Although there are more sophisticated formulations for the method [120, 52], we rely, as others [45], on the redundancy provided by extra light sources in order to avoid regions that deviate from the Lambertian assumption and from shadows. We use 5 white Luxeon® LED emitters arranged in a pentagon around the reference camera.

In our setup, the triangulation scanner and the photometric stereo scanner share the same reference camera. Therefore, normals and positions are automatically registered. Other hybrid scanner designs use independent cameras for depth and normal capture, usually acquiring normals at a higher resolution [8]. In these cases, after registration, depth data can be up-sampled to match the measured normal field, resulting in an output similar to that of our scanner.

4.3.2 Quality assessment

We need a way to assess the quality of measured normals and positions, and the accuracy of our results. Our triangulation scanner has been tested by scanning simple objects with known geometry, by comparing results with that of other scanners, and by analyzing how well different scans from the same object align together. We are confident that position errors are below 0.2mm.

One way to obtain ground truth for further comparisons is to produce a high-resolution range image with our triangulation scanner and use it as ground-truth when comparing to data obtained from much lower resolution input images. In practice, we compare full resolution scans with quarter resolution scans.

Following this idea, figures 4.5a and 4.5b present comparisons between the output of our scanners and ground truth for a closeup of the target object seen in figure 4.4. Position errors are defined as the distance to ground truth samples along lines of sight of the camera. Normal field errors give the angular deviation with respect to ground truth normals. To facilitate visualization, error values are mapped to colors.

Figure 4.5a shows the results for directly measured positions from our triangulation scanner. We can also compute normals from this geometry. Position errors in this example were below 0.5mm and can result from noise in the captured images, from speckles or from imperfections in the sub-pixel estimation of correspondences. The errors essentially take the form of random high-frequency noise throughout the object. Although relatively small, such errors can produce deviations in excess of 30° in the estimated normal field. When rendered with shading, these deviations are responsible for the distracting bumps seen in figure 4.1a.

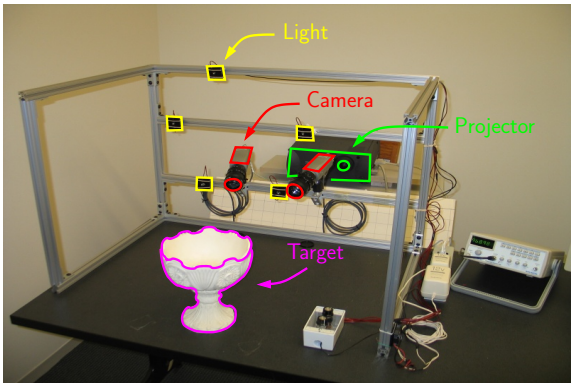
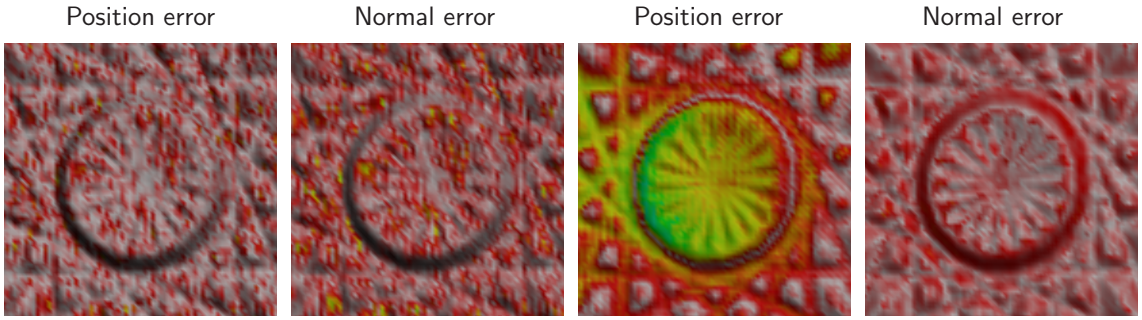
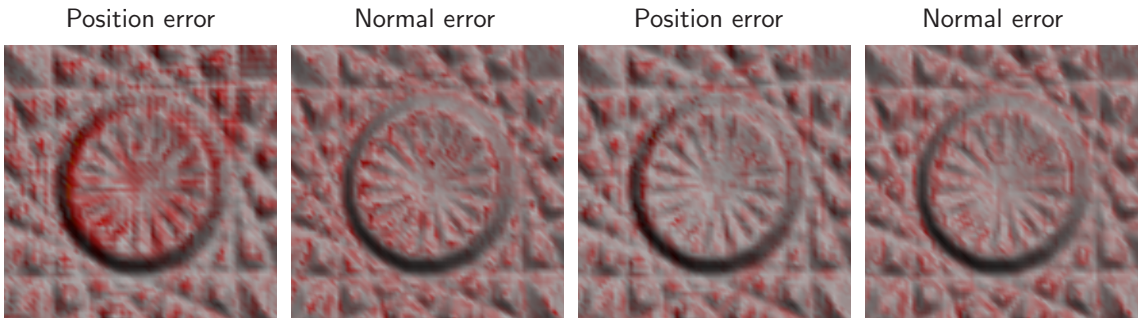


Figure 4.4: Hybrid scanner setup. For position measurement, two cameras capture synchronized images while a projector flashes random stripes at the object. For normal acquisition, one of the cameras takes pictures under illumination by 5 different, calibrated light sources. Part of this object is pictured in figures 4.2 and 4.5.



(a) Measured positions (b) Measured normals



(c) Corrected normals: section 4.4.1 (d) Optimization results: section 4.4.3

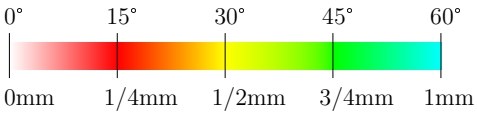


Figure 4.5: Quality assessment. (a-b) Notice how measured data contains considerable error. This takes the form of high-frequency noise for measured positions from a geometric scanner, and low-frequency bias for measured normals from a photometric scanner. (c-d) The combination of the two sources of data corrects most of the error present in the original data.

Figure 4.5b shows the results for normals measured by our photometric stereo scanner. Errors in measured normals are low-frequency in nature and often manifest themselves in terms of a systematic bias that can reach as much as 30°. These can be the result of interreflections, shadows, or of the oversimplification of the lighting model [104, 76]. When these normals are used to integrate for positions, large errors appear in the reconstructed surface.

4.4 Hybrid reconstruction algorithm

On the basis of the preceding error analysis, we conclude that the prevailing errors in measured normals, such as from photometric stereo, are low-frequency in nature, whereas measured positions, such as from a geometric depth scanner, contain mostly high-frequency noise. Therefore, our algorithm for combining these data sources considers frequency components independently. While it is possible to do this in a single step, we find that it is more efficient to proceed in two stages: first *correcting* the bias in the normals, then *optimizing* the geometry to conform to both the corrected normals and the measured positions.

Efficiency is one of our main design goals. Our normal correction step is very efficient and simply involves low-pass filtering and rotations. Combining the corrected normals and measured positions into an improved surface is more challenging. We formulate this as an optimization problem, with careful development of the objective function to enable solution as a sparse linear system. This allows our algorithm to operate efficiently on the large meshes typically found in computer graphics.

We start with the normal correction, then we describe an algorithm specialized to work on single range images, and finally we describe a method that operates on full models, i.e., arbitrarily tessellated triangle meshes.

4.4.1 Using positions to improve normals

A method to eliminate the bias in the measured normals that takes advantage of the underlying measured positions was presented by Rushmeier and Bernardini [104]. This method is specific to the photometric stereo setting, and therefore we developed a technique that can be applied to a wider range of input data.

As we have seen, the bias present in measured normals is low-frequency. On the other hand, the noise in normals computed from measured positions is high-frequency (compare figures 4.5a and 4.5b). By combining the appropriate frequency bands, we can obtain higher quality normal estimates.

Let N^p be the normal field indirectly computed from measured positions and let N^m be the directly measured normal field. Conceptually, we wish to replace the low-frequency component of N^m with data from N^p. We start by smoothing both fields by the same amount, which should be enough to eliminate the high-frequency noise present in N^p and the high-frequency detail present in N^m. The smoothing can be performed by individually convolving the coordinate functions of the normals with a Gaussian and then renormalizing (3D distances can be used instead of geodesic

distances). The resulting smoothed fields, $\mathrm{S}(N^p)$ and $\mathrm{S}(N^m)$, correspond to the low-frequency components of the original fields.

We then compute a rotation field R representing the rotations which move each normal in $\mathrm{S}(N^m)$ to the corresponding normal in N^m. Finally, we compute the corrected normal field $N^c = R\,\mathrm{S}(N^p)$ by applying the rotation field to the smoothed normal field obtained from measured positions. The rotation field captures the high-frequency detail in N^m, but is free of low-frequency information. Unlike vector addition, rotations transfer detail uniformly from one normal field to the other, regardless of the angular distance between the corresponding smoothed fields. Notice that since the normal field N^p is only used after severe smoothing, it can be obtained by virtually any method that produces normals from positions.

Figure 4.5c shows the resulting corrected normals and the improved positions obtained from their integration. We have eliminated the high-frequency noise in figure 4.5a, and significantly reduced the bias in figure 4.5b. There are still some errors, especially in terms of surface positions due to the inherent problems of integrating the surface normals without depth information. These will be addressed in the next subsection.

4.4.2 Using normals to improve positions

Our problem is to find, for each point, a new optimized position that conforms to the normals corrected by the method of section 4.4.1. We also would like to prevent these points from moving too far away from their original positions. Given the two objectives, we proceed by energy minimization, looking for the surface S that minimizes the weighted sum of two error terms, the *position error* E^p and the *normal error* E^n:

$$\arg\min_S \ \lambda E^p + (1-\lambda)E^n. \tag{4.1}$$

Here, the parameter $\lambda \in [0,1]$ controls how much influence the positions and normals have in the optimization. The two error terms are measured in units of squared distance, and therefore λ is dimensionless. When λ is 0, the algorithm considers normals exclusively, with help from measured positions only in boundary conditions, much like shape-from-shading (in fact, this is the method we use to *integrate* normals for comparison purposes in this paper). When λ is 1, the algorithm simply returns the original positions. For intermediate values, the method finds the optimal weighted combination of normals and positions.

We restrict our formulation to error terms that lead to sparse linear least squares minimization problems. After all, we are dealing with large problems that can contain hundreds of thousands, or even millions of variables. Efficiency both in time and space are therefore vital. In order to guarantee linearity, we restrict ourselves to terms in the form

$$\sum_i \left|\left|L_i(S) + c_i\right|\right|^2, \tag{4.2}$$

where the L_i are linear operators, each involving a subset of the variables defining the surface S, and the c_i are arbitrary scalar constants. To guarantee sparsity, we make sure that each L_i references only a small (i.e., constant) number of variables.

In this case, it is easy to see that the resulting optimization problem is equivalent to the solution, in the least squares sense, of the linear system

$$\begin{bmatrix} L_1 \\ L_2 \\ .. \\ L_n \end{bmatrix} S = \begin{bmatrix} c_1 \\ c_2 \\ .. \\ c_n \end{bmatrix}, \tag{4.3}$$

where L_i are rows defining each linear constraint, c_i are the associated constants, and S is a solution vector defining the optimal surface. This solution vector will contain depths in the range image case, and displacements in the normal direction for the full model formulation. Additionally, the method can be extended to use individual confidence estimates for each normal and position constraint by simply pre-multiplying the system by a diagonal weight matrix (i.e., using weighted least squares).

Another key feature in our formulation is that the optimization process commutes with the scaling, rotation, and translation of the input. By including only lengths and projected lengths in our error terms, we can express them as dot products. These are preserved by rigid body transformations, which leads to rotation and translation invariance. The scale invariance, on the other hand, comes from the fact that both the position and normal errors are scaled by the same amount, and therefore the optimal solution remains unchanged.

We have tested a variety of sparse solvers and all of them were able to deal with the linear systems that we formulate. Our first implementation was based on the Paige and Saunders [96] distribution of their Conjugated Gradient method for solving sparse least squares problems. We then moved to an implementation based on the PETSC library [84], which offers a variety of sparse preconditioners that improve the convergence of the iterative solver. Our latest implementation uses the direct solver of the CHOLMOD library [43], which is based on sparse Cholesky factorizations.

4.4.3 Range image formulation

Raw measured positions usually come organized in a range-image. The pixel coordinates on the reference camera induce a natural parametrization of the corresponding surface. Accordingly, under perspective projection, the coordinates of a surface point can be written in terms of a depth function $Z(x, y)$. In other words, given the pixel coordinates, the position of the corresponding surface point $P(x, y)$ has only one degree of freedom, $Z(x, y)$:

$$\begin{aligned} P(x,y) &= \begin{bmatrix} X(x,y) & Y(x,y) & Z(x,y) \end{bmatrix}^\top & (4.4) \\ &= \begin{bmatrix} -\frac{x}{f_x} Z(x,y) & -\frac{y}{f_y} Z(x,y) & Z(x,y) \end{bmatrix}^\top, & (4.5) \end{aligned}$$

where f_x and f_y are the camera focal lengths in pixels, and we have assumed the principal point to be at the origin.

We can define the position error as the sum of squared distances between the optimized positions and the measured positions:

$$E^p = \sum_i \left|\left|P_i - P_i^m\right|\right|^2, \tag{4.6}$$

where P_i is the i^{th} optimized position, and P_i^m is the corresponding measurement. To evaluate the position error as a function of depth only, we substitute equation 4.5 into equation 4.6 to obtain

$$\begin{aligned} \left|\left|P_i - P_i^m\right|\right|^2 &= \left(X_i - X_i^m\right)^2 + \left(Y_i - Y_i^m\right)^2 + \left(Z_i - Z_i^m\right)^2 & (4.7)\\ &= \left(\left(\frac{x_i}{f_x}\right)^2 + \left(\frac{y_i}{f_y}\right)^2 + 1\right)\left(Z_i - Z_i^m\right)^2 & (4.8)\\ &= \mu_i\left(Z_i - Z_i^m\right)^2, & (4.9) \end{aligned}$$

where $\mu_i = \left(\frac{x_i}{f_x}\right)^2 + \left(\frac{y_i}{f_y}\right)^2 + 1$ is a per-pixel constant that can be precomputed. It is clear that expression 4.9 is a linear term, in the form of expression 4.2.

The normal error could be defined in a number of different ways, including the sum of angular errors between corresponding normals in the optimized surface and the corrected normal field, or the sum of squared distances between each normalized or un-normalized pair. However, most formulations do not agree with expression 4.2 and lead to *non-linear* optimization problems. Our solution is to consider the tangents to the optimized surface instead. The corrected normals (which are constant) and the tangents to the optimized surface should be perpendicular. Recall that the surface tangents T_x and T_y at a given pixel can be written as linear functions of the depth values and their partial derivatives:

$$T_x = \frac{\partial P}{\partial x} = \begin{bmatrix} -\frac{1}{f_x}\left(x\frac{\partial Z}{\partial x} + Z\right) \\ -\frac{1}{f_y}\, y\frac{\partial Z}{\partial x} \\ \frac{\partial Z}{\partial x} \end{bmatrix}, \tag{4.10}$$

$$T_y = \frac{\partial P}{\partial y} = \begin{bmatrix} -\frac{1}{f_x}\, x\frac{\partial Z}{\partial y} \\ -\frac{1}{f_y}\left(y\frac{\partial Z}{\partial y} + Z\right) \\ \frac{\partial Z}{\partial y} \end{bmatrix}. \tag{4.11}$$

We now define

$$E^n = \sum_i \left[T_x(P_i)\cdot N_i^c\right]^2 + \left[T_y(P_i)\cdot N_i^c\right]^2, \tag{4.12}$$

where N_i^c is the corrected normal corresponding to P_i. E^n is the sum of squared projected lengths of the tangents to the optimized surface into the corrected normal field. It is minimized when all tangents are perpendicular to the corrected normals. To evaluate $T_x(P_i)$ and $T_y(P_i)$, we compute the partial derivatives of the depth function and substitute in equations 4.10 and 4.11. Since we are dealing with a uniform discrete sampling of the depth function, we can approximate the partial

derivatives by considering 3×3 neighborhoods and the following convolution kernels, assuming all neighbors are available:

$$\frac{\partial Z}{\partial x} = Z * \frac{1}{12} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -4 & 0 & 4 \\ \hline -1 & 0 & 1 \\ \hline \end{array}, \tag{4.13}$$

$$\frac{\partial Z}{\partial y} = Z * \frac{1}{12} \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -4 & -1 \\ \hline \end{array}. \tag{4.14}$$

Occasionally, around boundaries and depth discontinuities, some neighbors will not be present. We can detect these cases by analyzing the measured positions and use the best possible discrete derivative, down to simple one-sided derivatives. When there are no neighbors, the point can be removed from the minimization. In every case, the resulting term is still linear, in the form of expression 4.2.

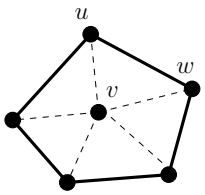
The optimal surface is then given by expression 4.1, which can be minimized by linear least squares. Each range image pixel generates at most 3 equations: one for the position error, and one for the normal error in each of the x and y directions. Fortunately, the matrix is sparse. In fact, the number of non-zero entries is *linear* in the number of pixels because there are at most 7 non-zero entries per row (one coefficient for the depth of the reference pixel and at most six others for the neighbors used to find the partial derivatives).

The resulting range image is accurate in both positions and normals, as the example in figure 4.5d shows. A variety of other examples are shown in figure 4.8.

4.4.4 Full model formulation

In some cases, we might be given a full 3D model with a normal map, and no access to the original range maps. This might be because the range maps never existed (e.g., a depth acquisition method that does not yield regularly-spaced range images was used), or because they are not available anymore. A generalization of the 2.5D method presented in the previous section to arbitrarily tessellated 3D meshes can be very useful in these cases.

In defining the new normal error $\hat{E}^n$, we no longer have a trivial way to compute the partial derivatives of the optimized surface at each vertex (needed to compute tangents), because there is no obvious parametrization. A solution that proved adequate is to consider the polygon formed by the neighbors of each vertex as an approximation to its tangent space. For each edge in each polygon, we add a term to the normal error that favors edges that are perpendicular to the measured normal at the central vertex:



$$\hat{E}^n = \sum_v \frac{1}{|v|} \sum_{u,w} \left[N_v^c \cdot (P_u - P_w) \right]^2. \tag{4.15}$$

Here u, w iterate over all edges in the 1-ring of v, and $|v|$ represents the number of such edges. N_v^c is the corrected normal at vertex v and is therefore constant. Hence, this formulation has the advantage of being linear. Furthermore, since small tangential vertex motions do not change the resulting surface significantly, we can restrict vertex motion to the corresponding corrected normal direction. This gives each vertex a single degree of freedom. To that end, let $P_i = P_i^m + \delta_i N_i^c$. The normal error assumes the form

$$\begin{aligned}
\hat{E}^n &= \sum_v \frac{1}{|v|} \sum_{u,w} \left[N_v^c \cdot (P_u - P_w) \right]^2 && (4.16)\\
&= \sum_v \frac{1}{|v|} \sum_{u,w} \left[N_v^c \cdot (P_u^m + \delta_u N_u^c - P_w^m - \delta_w N_w^c) \right]^2 && (4.17)\\
&= \sum_v \frac{1}{|v|} \sum_{u,w} \left[\delta_u \, N_v^c \cdot N_u^c - \delta_w \, N_v^c \cdot N_w^c + N_v^c \cdot (P_u^m - P_w^m) \right]^2 && (4.18)\\
&= \sum_v \frac{1}{|v|} \sum_{u,w} \left[a_{(v,u)} \delta_u - a_{(v,w)} \delta_w + b_{(v,u,w)} \right]^2, && (4.19)
\end{aligned}$$

where $a_{(u,v)} = N_u^c \cdot N_v^c$ and $b_{(v,u,w)} = N_v^c \cdot (P_u^m - P_w^m)$ are scalar constants that can be precomputed.

The position error assumes an even simpler form, reducing to

$$\hat{E}^p = \sum_i \left|\left| P_i^m - \left(P_i^m + \delta_i N_i^c \right) \right|\right|^2 = \sum_i \delta_i^2. \tag{4.20}$$

Both terms conform to expression 4.2 on the unknown δ_i, and therefore expression 4.1 can once again be minimized by linear least squares. The matrix is extremely sparse. In fact, each row has at most two non-zero coefficients. On a manifold mesh with n vertices, the average degree is 6, and therefore the number of equations is approximately $7n$ (n for $\hat{E}^p$ and $6n$ for $\hat{E}^n$). The problem can be solved just like the 2.5D version of the algorithm, and λ plays the same role.

The resulting full model has less high-frequency noise and more high-frequency detail than the simple merging of range scans. Examples of full model optimization are shown in figures 4.9, 4.10, and 4.11, and are discussed in the results section.

4.5 Results

In this section, we first evaluate the algorithm, discussing the setting of parameter values, and the accuracy and efficiency. We then discuss some applications, showing examples of some of the reconstructions we have produced using our method.

Accuracy and robustness: The precision of our method is one of its key features. The color-coded error renderings of figures 4.5c and 4.5d provide an indication of the quality of the results. Another way to assess the accuracy of our method is to analyze depth profiles for a reference object, as in figure 4.6. The plots show that measured positions are noisy and integration of corrected normals creates extraneous detail, whereas the optimized surface eliminates noise while closely following the ground truth.

Another important measure of surface precision is how well partial scans align with respect to each other. In a standard scanning pipeline, partial scans are first manually brought close to their aligned positions. Then, ICP [10] is run to precisely align scans pairwise. A global registration pass [101] is used to spread registration error evenly across pairwise alignments. Finally, all scans are merged into a single model using the VRIP volumetric method [38].

Good alignments are vital if details are to be preserved. In addition, since warped scans do not align properly, it is possible to use alignment quality to verify that scans are free of warp. Figure 4.7 compares our results against the alignment obtained by directly measured positions and by integration of photometric stereo normals. Low-frequency warps are clearly visible in the results of integration of normals. Conversely, high-frequency noise can be observed in the directly measured positions. By contrast, the results of our method are free of warp and align very well.

In general, our method is robust with regard to outliers both in positions and normal estimates. Position outliers can be easily detected and eliminated with the analysis of shapes and sizes of the triangles produced from the measured range images. Although harder to detect, the influence of normal outliers is limited to a small neighborhood.

Relative importance of positions and normals—setting λ: The λ parameter provides simple and effective control over the behavior of the algorithm, which consistently produces good results. Recall that λ is a dimensionless parameter in the range $[0, 1]$ and controls the relative contribution of normals and positions in the optimization. For instance, when the measured positions are noisy, a smaller λ, giving less weight to geometric data, will smooth the noise away. In general, we find that values in the range of $[0.1, 0.3]$ are most suitable for λ, and the method is not very sensitive to parameters in that range, with satisfactory results obtained for all the examples we tested. Values outside this range can be appropriate when the quality of positions and normals is less balanced.

Efficiency: One important characteristic of our method is its efficiency. By formulating only linear constraints, we are able to solve the minimization by linear least squares. Since each constraint refers to a constant number of variables, and the number of constraints is linear in the number of variables, the memory requirements are also linear. As the weight λ assigned to measured positions increases, the identity part of the linear system dominates (see equation 4.3) and convergence is greatly improved when using the LSQR method. Hence, optimizing a range image with $\lambda = 0$ (corresponding to integrating the surface normals directly) may require several minutes. However, using a value of λ within the suitable range [0.10, 0.30] allows us to obtain high quality results within seconds on models having hundreds of thousands of vertices. On the other hand,

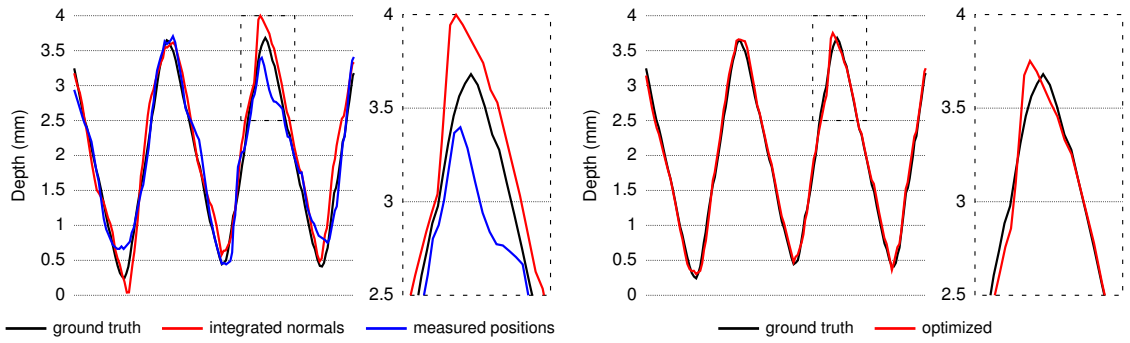


Figure 4.6: Depth profiles (in 0.5mm) for a reference object. (Left) Measured positions and integration from corrected normals, (Right) optimized surface. Note how the optimized profile follows the ground truth profile closer than the others.

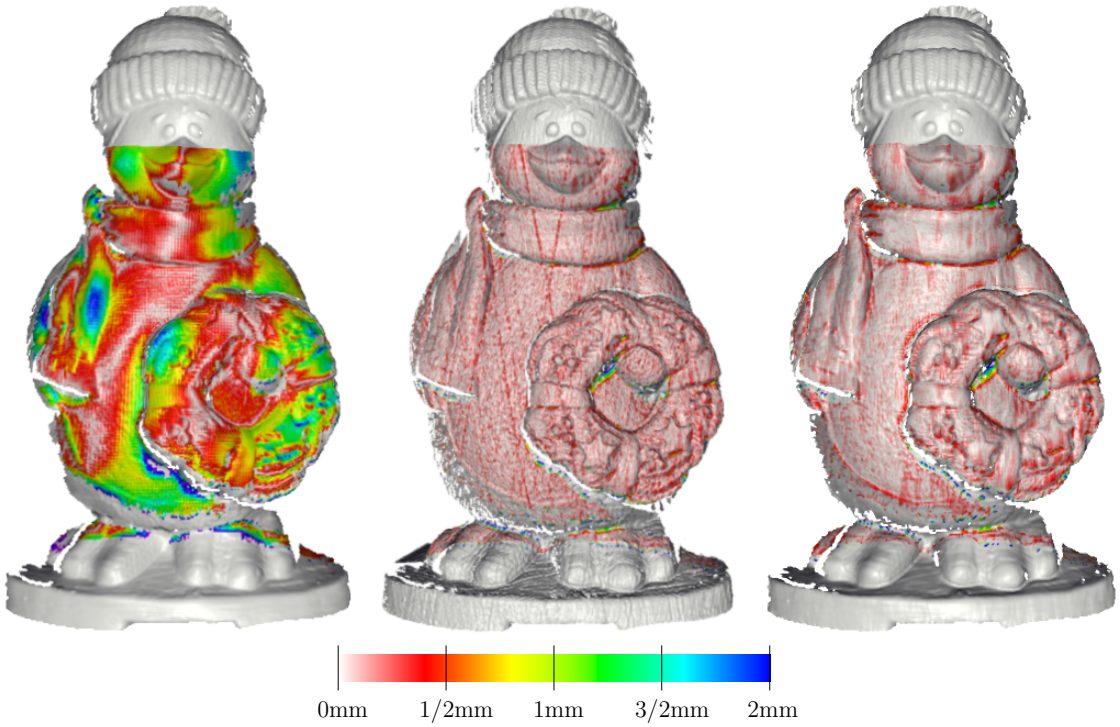


Figure 4.7: Distance between two aligned scans of a 30cm plaster penguin. (Left) Integration of normals, (Center) directly measured positions, (Right) optimized surfaces ($\lambda = 0.1$). Our method produces results that are free of warp and therefore align very well.

when using the PETSC library with preconditioning enabled, or when using the direct solver of CHOLMOD, the value of λ has little or no impact on the running times, which are always in the order of seconds.

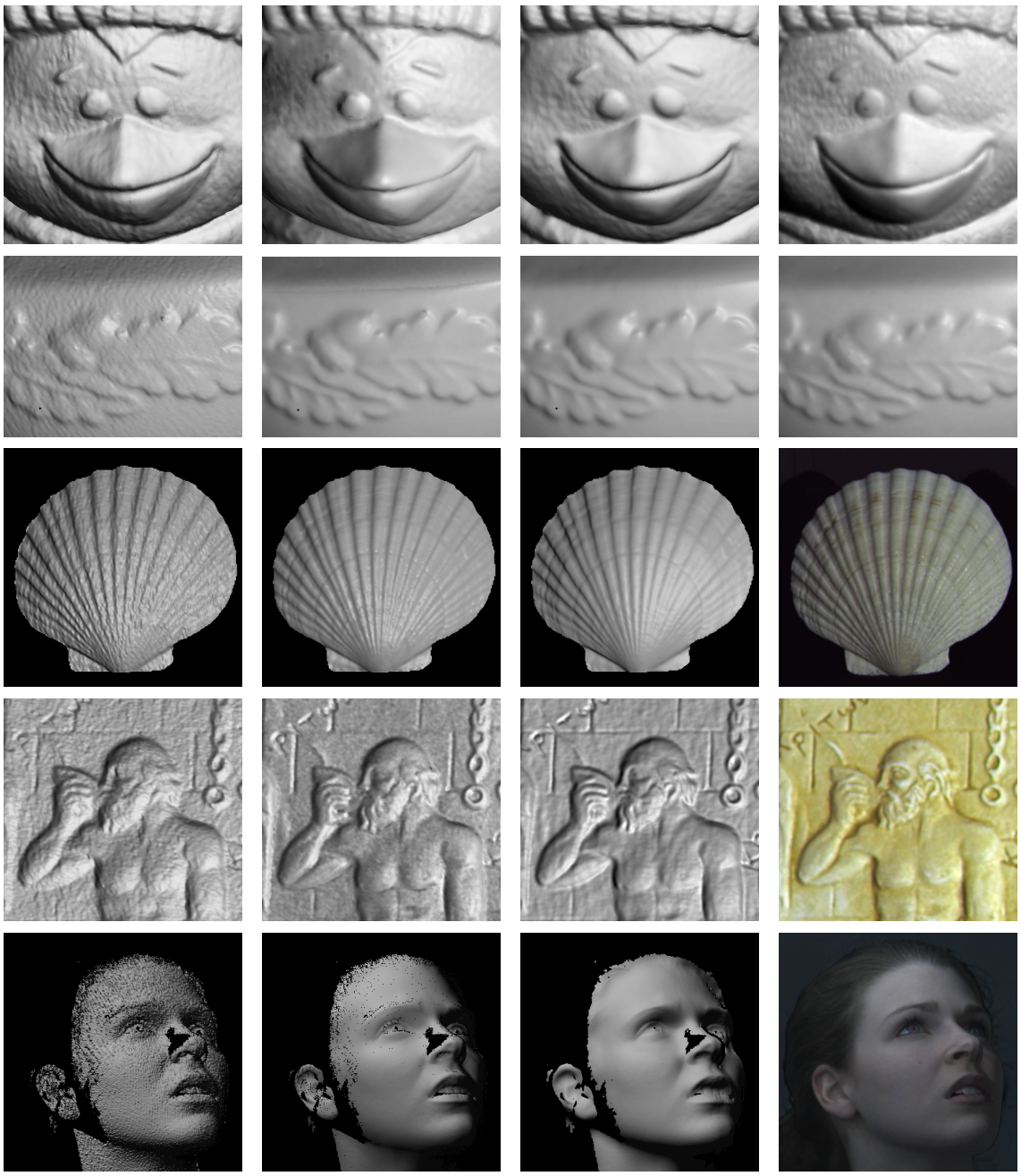
Range image optimization: Our method can produce very high-quality reconstructions of complex objects with hundreds of thousands of triangles, including sharp and high-frequency spatial features. Our results can be used for rendering, as well as other visualization and signal-processing tasks. In the course of this research, we have used our algorithm to create reconstructions of several objects, some of which are shown in figure 4.8. These were all optimized with the range image formulation of our error terms (section 4.4.3). The top row in figure 4.8 shows a closeup of 16 aligned scans of the plaster penguin used in figure 4.7. The second row shows a closeup of illustrations on a porcelain gravy boat. The third row shows a model obtained from a sea shell. The fourth row is a copy of figure 4.1, for completeness. The last row shows a scan of a human face, courtesy of Jones et al. [66] and the USC Institute for Creative Technologies.

As discussed in the introduction, errors in the normals estimated from the scanned geometry produce unpleasant bumps (a). Using the normals acquired by photometric stereo (b) eliminates the bumps, but introduces bias. The optimized geometry (c), on the other hand, is almost free of noise or bias. Furthermore, we obtain accurate renderings at grazing angles, and with mesh operations like accessibility shading (figure 4.2). Indeed, many rendering tasks like ray-tracing reflections, silhouette computations, suggestive contours, and lighting will benefit from the noise free, unbiased normals of our method.

Full model optimization: We present three results of the full model version of our algorithm (subsection 4.4.4). The model shown in the left part of figure 4.9 is the result of the alignment and merging of 15 raw range scans into a full model. Normals for the same object were measured by another 15 photometric stereo scans. These normals were then corrected for bias (as in section 4.4.1) and mapped to the vertices of the merged model. Finally, the vertex positions were then optimized to conform to the mapped normal constraints. The result, shown in the right of figure 4.9, clearly has less high-frequency noise and more detail.

As an example of application of our method to unconventional sources of surface normal and geometry information, consider the quarter in figure 4.10. The model was produced by the mesh optimization of flat geometry with a bump map. The flat geometry was generated procedurally, from the specification of the coin dimensions. The bump maps for the two sides of the coin were produced with a flat-bed scanner. The bump map for the edge was produced procedurally from the specification of the number of ridges. The figure shows a model lit by normals computed from the optimized geometry.

Finally, the dataset shown on figure 4.11 was courtesy of Weyrich et al. [124] and the Mitsubishi Electric Research Laboratories. The output of their triangulation scanner consists two range images merged into a single face model. The setup also captures a registered normal field, which we used to improve the geometry. The results allowed their group to investigate the fine surface detail of a large collection of human faces.

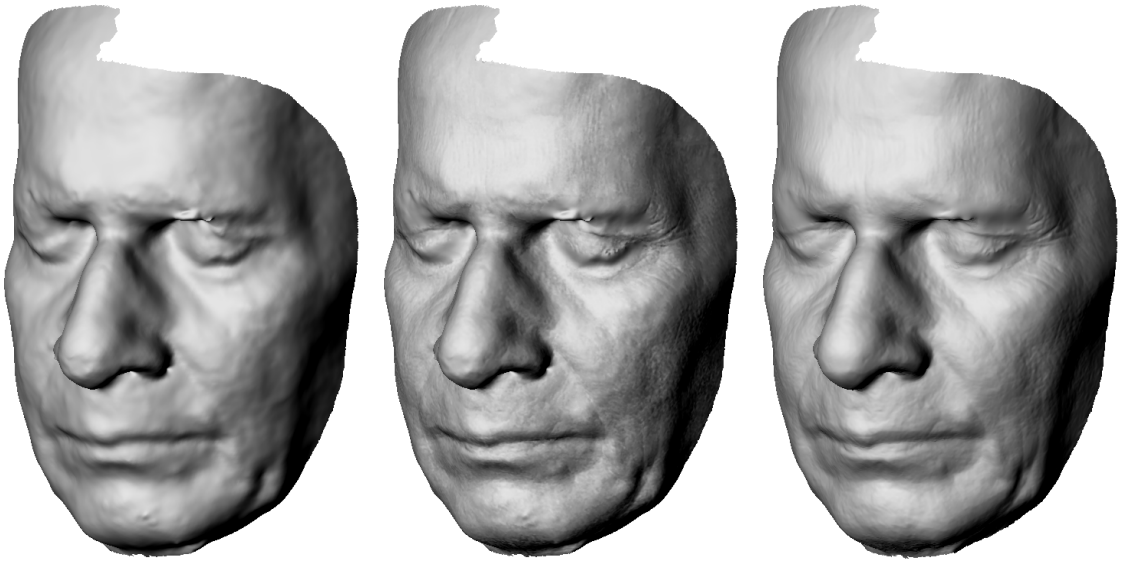


(a) (b) (c) (d)

Figure 4.8: Rendering comparisons. (a) Rendering of 3D scanned range image, (b) same scanned geometry, augmented with a measured normal-map (from photometric stereo), (c) our hybrid surface reconstruction, (d) photograph. The first row is the result of aligning several range images for the penguin shown in figure 4.7. The second row shows a closeup of a porcelain gravy boat. The third row shows scans of a sea shell. The fourth row shows scans of a plaster replica of a Greek panel. The last row shows scans of human face (the dataset was courtesy of Jones et al. [66] and the USC Institute For Creative Technologies).

Figure 4.9: (Left) Several range scans were aligned and merged. (Right) Normals coming from photometric stereo were mapped to the merged model, which was then optimized with your method. Notice how the noisy geometry was improved.

Figure 4.10: The figure shows an example with alternative sources for geometry and normals. A model of a quarter was produced from initially flat geometry and a bump map. The figure shows optimized model with normals recomputed from geometry.



(a) (b) (c)

Figure 4.11: The scanner that produced the model in (a) does not return range images. Instead, it outputs the result of merging two range images. A light dome was used to capture registered normals (b). The data was later optimized using our algorithm. Notice how the detail was transfered from the normal field to the geometry (c). The dataset was courtesy of Weyrich et al. [124] and the Mitsubishi Electric Research Laboratories.

4.6 Conclusions

3D geometry is commonplace in computer graphics. Typically, this information comes in the form of depth and vertex positions, or normals and surface orientations. However, neither depth information alone, normal information alone, nor a simple combination such as a bump map provide a complete solution for computer graphics modeling and rendering. In this chapter, we have shown that when we have access to positions and normals for a 3D model, it is possible to combine these two sources of information into an optimal surface. If only the most reliable component of each source of information is considered, the resulting surface will be more precise than that obtainable by each source independently. We presented an analysis of the common error characteristics of standard position and normal acquisition systems, and designed our algorithm to account for these types of errors. By formulating the problem in a particular way, we reduced it to the solution of a sparse linear system, enabling very efficient computation of optimal surfaces for large meshes. Our algorithm represents the first practical technique in computer graphics for combining position and normal information into a precise surface reconstruction.

Chapter 5

Specularity triangulation

A popular class of systems for determining the 3D shape of real-world objects relies on the notion of multiview consistency. These systems may be thought of as hypothesizing 3D points, then evaluating whether their projections into two or more viewpoints are consistent with images taken from those views. In the simplest case, the consistency is evaluated based on color. More sophisticated systems might evaluate the consistency of windows of pixels (as is the case in many stereo systems), or might consider temporal variation (as in temporal active stereo or structured light systems). Any of these consistency criteria may be used to construct a matching cost function, which is then used to perform triangulation to find geometry (possibly with an additional stage that may enforce smoothness of matches etc.). With few exceptions, however, such systems have focused on reconstructing diffuse objects and scenes.

In this chapter, we consider the problem of reconstructing specular objects, positioned in a scene or environment of known geometry and appearance. Because the images of these objects consist entirely of reflections of the scene around them, simple consistency metrics based on color, pixel windows, or temporal variation will not be effective. Instead, we rely on a *specular consistency criterion* that considers the reflections of the scene visible in the object. This specularity consistency criterion must therefore operate on position/normal tuples, rather than on positions alone. To evaluate whether a proposed position/normal exhibits specularity consistency with respect to one image, we reflect the ray from the camera to the given point off a surface patch having the given normal. The consistency criterion is that the intersection of the reflected ray with the known scene is consistent with the pixel observed in the camera image.

Given only a single camera position, we find that an infinity of position/normal tuples will necessarily satisfy the consistency condition: for any proposed point, we can find a normal such that the reflected ray hits any desired scene point. Two camera positions, however, provide disambiguating information, and in most cases restrict us to a single allowed position/normal (exceptions are discussed later in the chapter). Note that, in general, a point is reconstructed not because different cameras observe the same part of the scene reflected from it: they observe different reflections consistent with the hypothesized position/normal.

The specularity consistency condition proposed above has been previously identified and exploited in other contexts [108, 19]. We propose to use the constraint to define a matching cost function for two camera views, and demonstrate that the condition may be used for dense stereo reconstruction of specular 3D objects. We analyze our system on synthetic imagery, and show real-world results. Our contributions include:

- defining a stereo matching cost function based on specularity consistency;
- proposing a novel normal-based anisotropic diffusion scheme for the matching cost, which strengthens matches lying on a continuous surface;
- presenting a theoretical analysis of the ambiguities that can arise from the specularity consistency;
- demonstrating dense and accurate 3D reconstruction of specular objects, in a setup in which the "known scene" consists of a computer monitor displaying temporally-varying patterns.

This chapter presents unpublished work resulting from a collaboration with Tim Weyrich and Szymon Rusinkiewicz.

5.2 Related work

Specular surfaces have been widely studied in the literature. By imposing continuity and smoothness constraints, depth can be indirectly obtained from normal estimates [56, 117]. In the case of glossy surfaces, normal estimates can be computed for example by fitting of explicit or parametric models [61, 32, 88] or by direct detection of highlights [30].

On the other hand, most work on stereo reconstruction in the presence of specularities attempts to avoid view dependent effects. Approaches that cope with specular highlights include capturing multiple views [11, 79], masking them out with polarization filters [125, 89], treating them as occlusions [78], or removing them from the captured images [114, 79].

Naturally, for the class of objects we are considering (i.e., smooth, glossy materials), view-dependent information provides the most useful of constraints. A wide range of methods derive local shape information from the identification and/or tracking of the distorted reflections of light-sources and special known features [14, 15, 135, 95, 109]. These methods tend to produce sparse reconstructions.

Dense measurements can be produced, for example, by the general framework of light-path triangulation [74]. Within this framework, our method can be described as a novel technique for $\langle 2, 1, 1\rangle$-triangulation. Conversely, Bonfort et al. [20] present a method for $\langle 1, 1, 2\rangle$-triangulation that requires just one viewpoint, but two known reference points per ray. Outside of this framework, shape has been recovered by rotating an object under known illumination [132], and by exploring the Helmholtz reciprocity [134].

The idea of checking for normal consistency across different views has been originally proposed by Sanderson et al. [108], in the context of feature matching of specular highlights, and later used by Bonfort and Sturm [19], as part of a voxel carving method for recovering the shape of

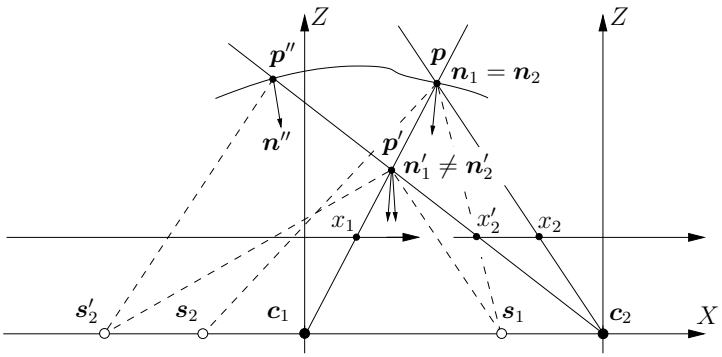


Figure 5.1: The specularity consistency. In order to find the pixel x_2 in camera 2 that corresponds to the image x_1 of a point $\boldsymbol{p}$ as seen by camera 1, we compare the normal direction hypotheses formed by the light source positions $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ that produce highlights through x_1 and x_2, respectively. An incorrect match x_2' is likely to produce normal hypotheses $\boldsymbol{n}_1'$ and $\boldsymbol{n}_2'$ that do not agree, which allows us to reject the match. In contrast, the correct match will always result in consistent normal estimates by the two cameras.

specular objects. In contrast, by rephrasing this constraint as a matching cost function, we can leverage decades of research on stereo reconstruction [110]. In particular, we can take advantage of efficient global matching algorithms that produce satisfactory dense reconstructions even when the matching cost functions are not particularly discriminant.

5.3 Triangulation by specularity consistency

The reconstruction method we propose relies on the ability to densely identify environmental scene points reflected by a specular surface. This can be achieved by illuminating the specular object with a dense set of controllable light sources. Various setups have been proposed to create a temporally encoded lighting environment using monitor pixels as light sources [136, 121, 20]. Using a suitable encoding scheme, the source position $\boldsymbol{s}$ of light reflected by a specular surface point $\boldsymbol{p}$ toward a camera can be decoded from the camera's intensity observations over time. In section 5.6, we present a similar prototype system to evaluate our approach.

With the light source position $\boldsymbol{s}$ known, the normal $\boldsymbol{n}$ at a point $\boldsymbol{p} \in \mathbb{R}^3$ can easily be computed as the bisector

$$\boldsymbol{n} = \frac{\boldsymbol{l} + \boldsymbol{v}}{\|\boldsymbol{l} + \boldsymbol{v}\|} \tag{5.1}$$

between lighting and viewing directions

$$\boldsymbol{l} = \frac{\boldsymbol{s} - \boldsymbol{p}}{\|\boldsymbol{s} - \boldsymbol{p}\|} \quad \text{and} \quad \boldsymbol{v} = \frac{\boldsymbol{c} - \boldsymbol{p}}{\|\boldsymbol{c} - \boldsymbol{p}\|} \,, \tag{5.2}$$

with $\boldsymbol{c}$ the center of projection of the observing camera. In the context of stereo reconstruction, however, depths and hence surface point positions are not known in advance. Instead, a *candidate*

point $\boldsymbol{p}$ corresponding to a disparity hypothesis has to be tested for consistency.

Consider figure 5.1. A point $\boldsymbol{p}$ on the specular surface projects to pixel x_1 as seen by camera 1. Assume we also know the light source position $\boldsymbol{s}_1$ that casts a highlight through x_1. We want to find the corresponding pixel x_2 to which $\boldsymbol{p}$ projects at camera 2, so we can triangulate for its position. Naturally, we must be able to distinguish the wrong candidate matches x_2' from the correct match x_2.

The figure shows a sample incorrect match x_2'. By triangulation, this match determines an incorrect position $\boldsymbol{p}'$ along the line of sight through x_1. Given $\boldsymbol{p}'$ and $\boldsymbol{s}_1$, camera 1 hypothesizes a normal direction $\boldsymbol{n}_1'$ (equation 5.1). Camera 2, on the other hand, makes an independent normal direction hypothesis $\boldsymbol{n}_2'$. This hypothesis comes from $\boldsymbol{p}'$ and the light source position $\boldsymbol{s}_2'$ that casts a highlight through x_2'. Recall that, in the figure, $\boldsymbol{p}$ does not project to x_2'. Some other surface point $\boldsymbol{p}''$ does, with corresponding normal direction $\boldsymbol{n}''$. This is the point responsible for the light source position $\boldsymbol{s}_2'$ that is visible through x_2'. The independence between points $\boldsymbol{p}$ and $\boldsymbol{p}''$ is likely to cause the normal direction hypotheses to disagree, in other words $\boldsymbol{n}_1' \neq \boldsymbol{n}_2'$.

In contrast, the correct correspondence x_2 actually images the point $\boldsymbol{p}$. Triangulation produces the correct depth, and the light source position $\boldsymbol{s}_2$ visible through x_2 is in fact reflected from $\boldsymbol{p}$. Since both x_1 and x_2 observe the same surface point, the normal direction hypotheses $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ will agree. We present a framework for dense stereo reconstruction that exploits this fact.

5.4 Dense stereo framework

Dense stereo reconstruction generally requires a metric that assesses a disparity hypothesis d for a given image point (x, y). This is typically expressed by assigning a matching cost value $C(x, y, d)$ to this hypothesis. A dense stereo algorithm then aims at assigning minimum-cost disparities to all pixels in a reference image, for instance, the image captured by camera 1. Algorithms differ in whether disparity minimization takes place locally or globally, and in how an overall minimum is defined and computed [110]. However, matching costs are universal, and arbitrary cost functions can be directly used with most existing algorithms.

5.4.1 Matching Costs

Stereo matching cost computation is usually based on image intensities and considers squared or absolute differences between corresponding pixels in the image pairs. That is, the cost of matching pixel (x, y) in image 1 with a pixel $(x + d, y)$ in image 2 can be expressed as

$$C(x, y, d) = ||I_1(x, y) - I_2(x + d, y)|| \,, \tag{5.3}$$

with $I_i(x, y)$ denoting the intensity at a pixel (x, y) in the image captured by camera i.

In contrast to this, our reconstruction framework is based on differences in normal hypotheses rather than intensities. A given disparity hypothesis d for a pixel (x, y) corresponds to a candidate point $\boldsymbol{p}$ in space. As explained in section 5.3, $\boldsymbol{p}$ produces two normal estimates $\boldsymbol{n}_i(\boldsymbol{p})$ from the two

cameras. We use the angular difference

$$\delta(x, y, d) = \cos^{-1}\big(\boldsymbol{n}_1(\boldsymbol{p})^\top \boldsymbol{n}_2(\boldsymbol{p})\big) \tag{5.4}$$

between the normal estimates as a correspondence measure for a hypothesis (x, y, d). If $\boldsymbol{p}$ exactly coincides with a specular surface, δ is expected to be zero. In a realistic setting, however, there will be a slight error in the normal estimate due to noise and to pixel- and light-source quantization, that is, the measured δ will deviate from the ideal δ_0 by $\delta = \delta_0 + \delta^*$. Assuming the error term δ^* to be mean-free and normal-distributed with standard deviation σ, the likelihood of $\delta(x, y, d)$ denoting a match is

$$P(x, y, d) = \frac{1}{\sigma\sqrt{2\pi}}\, \mathrm{e}^{-\frac{\delta(x,y,d)^2}{2\sigma^2}} \ . \tag{5.5}$$

In our experiments, we choose σ to be within 5° and 8°, depending on the data quality. From P we set the matching cost function to be

$$C(x, y, d) = 1 - P(x, y, d) \ . \tag{5.6}$$

This cost function can now be used with a variety of existing stereo reconstruction algorithms. Section 5.7 shows example reconstructions using different alternatives.

5.4.2 Normal-aware cost aggregation

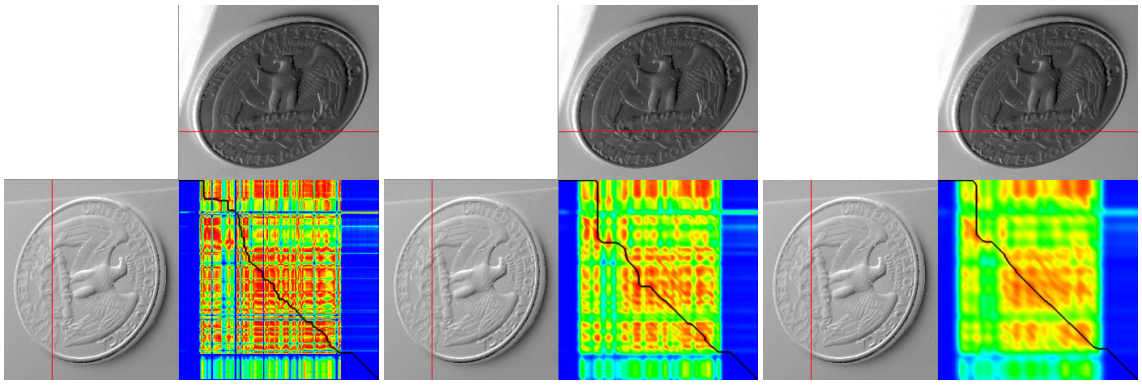
In traditional (intensity-based) stereo, the cost computation is often extended to a window region around the point of interest. This increases the reliability of the cost assessment for textured regions. Cost aggregation within a window can be expressed as a 2D or 3D convolution

$$C'(x, y, d) = (w * C)(x, y, d) \tag{5.7}$$

with a window function w [110]. This convolution can alternatively be expressed as a possibly anisotropic diffusion process [112].

When w is anisotropic, it introduces a bias for a certain slope during the surface reconstruction stage. In particular, if w is a 2-dimensional kernel over x and y (a common choice), the convolution corresponds to the integration over a window in the image domain, introducing a bias for image-parallel surface slopes.

We exploit the additional knowledge of normal directions to specifically bias the surface reconstruction toward the surface orientation corresponding to the normals. The intersection of the object surface with the epipolar (xd-)plane of y corresponds to a minimum-cost ridge in the cost function $C(x, y, d)$. Any surface reconstruction algorithm has to trace this ridge. In order to improve the precondition of the surface reconstruction, we perform anisotropic diffusion of the symmetrized cost function $F_y(x_1, x_2) \equiv C(x_1, y, x_2 - x_1)$, aggregating costs along the ridge direction predicted by the normal estimates. As we have shown in equation 3.6 of section 3.2, given



Source F_y | F'_y after 4 iterations | F'_y after 16 iterations

Figure 5.2: Consistency values within an epipolar plane (red line) through a problematic region with many false matches. High confidence values are displayed in red; the black line shows the result of a disparity optimization using dynamic programming. Anisotropic diffusion according to the normal estimates attenuates false positives. The images show (from left to right) unmodified consistency values $F_y(x_1, x_2)$ and $F'_y(x_1, x_2)$ after 4 and 16 diffusion steps, respectively.

a pair of rectified cameras, the unit tangent direction $\boldsymbol{t}$ to the matching ridge at (x_1, x_2) can be expressed as

$$\boldsymbol{t} \propto (\boldsymbol{n}^\top \boldsymbol{p}_1, \boldsymbol{n}^\top \boldsymbol{p}_2)^\top \tag{5.8}$$

where $\boldsymbol{n}$ and $\boldsymbol{p}_i$ are, respectively, the 2D projections of the normal and position of the surface point at (x_1, x_2), on the epipolar plane of y relative to camera i.

Following equation 5.7, we convolve F_y by a spatially-varying oriented Gaussian filter kernel g:

$$F'_y(x_1, x_2) = (g * F_y)(x_1, x_2) \;, \tag{5.9}$$

with

$$g(u, v) = \mathrm{e}^{-\boldsymbol{x}^\top \boldsymbol{V}^{-1} \boldsymbol{x}}, \quad \boldsymbol{x} = (u, v)^\top \;, \tag{5.10}$$

controlled by a spatially-varying variance matrix $\boldsymbol{V} \in \mathbb{R}^{2\times 2}$ dependent on the location of the filter application (x_1, x_2):

$$\boldsymbol{V} = r_{\mathrm{aa}}\mathbf{I} \;+\; r_{\mathrm{d}} \sum_{i\in\{1,2\}} \boldsymbol{t}_i \boldsymbol{t}_i^\top + F_y(x_1, x_2) \bar{\boldsymbol{t}}_i \bar{\boldsymbol{t}}_i^\top \;, \tag{5.11}$$

with $\boldsymbol{t}_i$ the unit vector in direction of the slope according to the normal estimate $\boldsymbol{n}_i$ and $\bar{\boldsymbol{t}}_i$ unit length and orthogonal to $\boldsymbol{t}_i$. Each summand after the summation sign implements the variance matrix of an oriented Gaussian along $\boldsymbol{t}_i$, each converging to an isotropic Gaussian as the local matching cost reaches one. Adding the variance matrices corresponds to convolving the respective Gaussians. The additional summand $r_{\mathrm{aa}}\mathbf{I}$ provides an anti-aliasing prefilter to guarantee a faithful

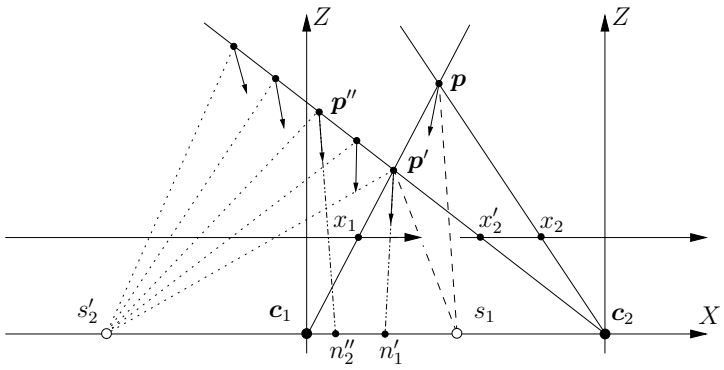


Figure 5.3: Weak ambiguities. Given a surface point $\boldsymbol{p}$, observed as (x_1, s_1) by camera 1, and a false candidate match x'_2, the surface cannot go through the intersection point $\boldsymbol{p}'$. However, for each other point $\boldsymbol{p}''$ along the ray through x'_2, there is a surface orientation that results in an observation (x'_2, s'_2), consistent with (x_1, s_1).

discretization of g. In our implementation we use $r_{\text{aa}} = 0.4$ and $r_{\text{d}} = 1.41$. Finally, in order to ensure energy preservation, each filter kernel is normalized after discretization to add up to one.

By iteratively applying equation 5.9, minimum-cost ridges are emphasized where both normal estimates agree with the matching ridge, while spurious ridges are attenuated. Figure 5.2 shows an example of problematic epipolar plane through the quarter dataset: the complex leaf pattern on the coin, and the non-specular background object introduce many spurious matches that are gradually smoothed out by the diffusion process.

5.5 Ambiguities

Even disregarding interreflections, the specularity constraint can lead to ambiguities. An observation by camera i can be described by a pair (x_i, s_i), where x_i is the pixel coordinate and s_i the light source position that produces a highlight through x_i. Here, for simplicity, and without loss of generality, we assume that the light source lies along the baseline between the two cameras (as in figure 5.3). Ideally, for any surface, and for each observation (x_1, s_1) by camera 1, there would exist only *one* corresponding consistent observation (x_2, s_2) by camera 2. Unfortunately, this is not the case. We can distinguish between two types of ambiguity. A *week ambiguity* occurs when at least two different observations, (x_2, s_2) and (x'_2, s'_2), are consistent with (x_1, s_2). A *strong ambiguity* occurs when *every* observation (x_2, s_2) is consistent with (x_1, s_1).

5.5.1 Weak ambiguities

Consider a pixel x'_2 from camera 2 that does not correspond to x_1, as shown in figure 5.3. By triangulation, we can determine the position $\boldsymbol{p}'$ for a (non-existent) hypothetical surface point:

$$\boldsymbol{p}'(x_1, x'_2) = (X', Z')^\top = \left(\frac{x_1 T}{x_1 - x'_2}, \frac{T}{x_1 - x'_2}\right)^\top. \tag{5.12}$$

From $\boldsymbol{p}'$ and s_1, we can compute the normal direction expected by camera 1 at $\boldsymbol{p}'$. This normal direction is uniquely determined by the intersection point n_1' of the normal ray at point $\boldsymbol{p}'$ with the baseline between the two cameras. To find the intersection, we use the angle bisector theorem on the triangle formed by points $\boldsymbol{p}'$, $\boldsymbol{c}_1$, and s_1:

$$n_1'(x_1, s_1, x_2) = \frac{s_1\sqrt{X'^2 + Z'^2}}{\sqrt{X'^2 + Z'^2} + \sqrt{(X' - s_1)^2 + Z'^2}}. \tag{5.13}$$

If (x_2', s_2') is consistent with (x_1, s_1), then the light from s_2' must reflect at $\boldsymbol{p}'$ towards the ray through x_2'. Using $\boldsymbol{p}'$ and n_1', and the angle bisector theorem on the triangle formed by points $\boldsymbol{p}'$, $\boldsymbol{c}_2$, and s_2', we can compute:

$$s_2'(x_1, s_1, x_2') = -\frac{n_1'^2(T - 2X') + (T - 2n_1')(X'^2 + Z'^2)}{n_1'^2 + 2T(X' - n_1') - (X'^2 + Z'^2)}. \tag{5.14}$$

Even though the ray through x_2' does not hit a surface point at $\boldsymbol{p}'$ (this point would occlude $\boldsymbol{p}$, which is by assumption visible through x_1), the ray may intersect the surface at *another depth* Z''. For each possible point $p'' = (X'', Z'')^\top = (x_2'Z'' + T, Z'')^\top$ along the ray through x_2, we can calculate a normal direction that reflects s_2' towards x_2'. Once again, we use the angle bisector theorem on the triangle formed by points $\boldsymbol{p}''$, $\boldsymbol{c}_2$, and s_2':

$$n_2''(x_1, s_1, x_2', Z'') = \frac{T\sqrt{(s_2 - X'')^2 + Z''^2} + s_2\sqrt{(X'' - T)^2 + Z''^2}}{\sqrt{(s_2 - X'')^2 + Z''^2} + \sqrt{(X'' - T)^2 + Z''^2}}. \tag{5.15}$$

In summary, the rays through x_1 and x_2' observe independent surface points, and intersect at a virtual point $\boldsymbol{p}'$ (equation 5.12). Camera 1 expects $\boldsymbol{p}'$ to have normal n_1' (equation 5.13). For each x_2', we can compute a light source position s_2' that causes (x_2', s_2') to be consistent with (x_1, s_1) (equation 5.14). Then, for each point $\boldsymbol{p}''$ along the ray through x_2', we can compute a normal direction n_2'' that reflects light from s_2' towards the ray through x_2' (equation 5.15). Whenever the surface goes through a point $\boldsymbol{p}''$ with the appropriate normal direction n_2'', we have a weak ambiguity.

5.5.2 Strong ambiguities

Consider figure 5.4. A point $\boldsymbol{p}$ generates an observation (x_1, s_1) by camera 1. Take now an arbitrary point $\boldsymbol{p}' = (X', Y')^\top$ being observed by camera 2 through pixel x_2'. The ray through x_2' intersects the ray through x_1 at a point that we can obtain from equation 5.12. Following equations 5.14 and 5.15, we can obtain a normal direction at $\boldsymbol{p}'$ that makes the observation at x_2' consistent with the observation at x_1.

In other words, each observation by camera 1 imposes a normal field on the XZ-plane. Starting from any point in the XZ-plane (i.e., an initial condition), any trajectory that respects this normal field (i.e., solves the associated differential equation) will generate a strong ambiguity curve. The

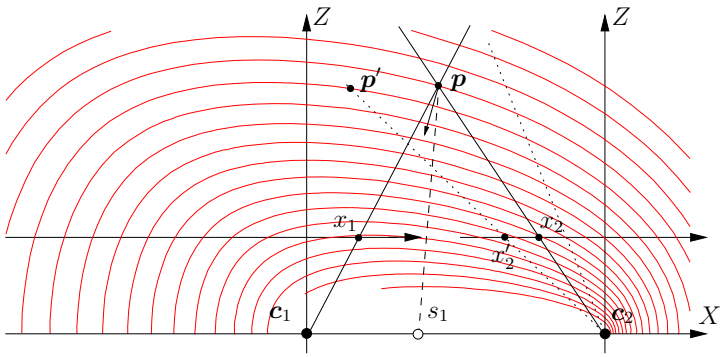


Figure 5.4: Strong ambiguity curves. An observation (x_1, s_1) by camera 1 imposes a normal field on the XZ-plane. If a surface conforms to this normal field, every observation (x_2, s_2) by camera 2 will be consistent with (x_1, s_1). In other words, we have a strong ambiguity.

curves in figure 5.4 were generated by numerical integration. If the surface follows one of these curves, *every* observation by camera 2 will be consistent with (x_1, s_1) by camera 1. These are the strong ambiguity curves.

In theory, weak and strong ambiguities are extremely unlikely. In practice, finite precision on the light source position estimation, as well as noise and discretization artifacts allow the phenomenon to manifest itself as wrong matches. These can be particularly problematic if the surface locally approximates a strong ambiguity curve.

5.6 Acquisition

Specular triangulation requires a camera pair and an apparatus to illuminate an object with a dense set of light sources. To that end, the cameras acquire images of the object being lit by a temporally encoded light pattern. In principle, any light pattern that allows for the derivation of the identity of a single light source L_i in a specular reflection observed by a camera at $\boldsymbol{c}_i$ can be used.

Efficient encoding schemes, such as gray-code patterns, exist; however, specular reflection is subject to a convolution with a specular lobe. Accordingly, the chosen pattern has to be robust under convolution. A very robust encoding would be to trigger one light source at a time. For each surface point, the light source $L^i_{\max}$ is determined that produces a maximum intensity response under observation from camera i. The temporal location of the maximum response is comparatively stable even under convolution with a specular lobe. For a dense set of many light sources, however, this procedure is highly inefficient.

In order to reduce acquisition times, we propose to use multiple linear light sources instead of a dense set of single point light sources. During acquisition, the linear light sources are swept through space [51]. For each surface point, the maximum response during each sweep is determined. In a good approximation, these maxima occur when the linear light source covers $L^i_{\max}$. Accordingly, we intersect the locations of the corresponding illuminating lines to obtain the position $\boldsymbol{s}_i$ of $L^i_{\max}$.

5.6.1 Acquisition Procedure

As a practical setup, we realize the linear light source sweeps using an LCD monitor displaying white stripes at different positions and orientations. In this paper, we present simulated and measured results for a 24" monitor that implements two line sweeps by displaying shifting horizontal and vertical stripes, respectively. The stripes are 0.5 cm wide and are subsequently shifted by one stripe width. Scanning times can be improved by multiplexing the stripes according to a Hadamard pattern [113], but we gave preference to quality over speed in the acquisition system, in order to avoid additional sources of error. Figure 5.5 shows our experimental setup. Both the camera pair and the monitor's location with respect to the cameras are calibrated. Knowing the light source locations $\boldsymbol{s}_i$, specular triangulation can be performed as described in section 5.3.

In practice, the crucial part is a reliable estimation of $L^i_{\max}$ from the intensity variation of each observed point over time. Camera noise and the discretization of the line sweep prevent us from directly picking the stripe location with the maximum intensity response. Instead, we produced good results by fitting a Gaussian mixture model

$$I(t) = \sum_i^n a_i \mathrm{e}^{-\frac{(t-\mu_i)^2}{\sigma_i^2}} \tag{5.16}$$

to the data, and deriving the maximum intensity point in time as μ_i of the narrowest Gaussian lobe, that is, for i with minimum σ_i. The intention is to allow the fit to approximate diffuse and ambient reflectance contributions by wider lobes, and to model the specular peak by a single, narrow lobe.

A common choice for fits of Gaussian mixture models is the Expectation Maximization algorithm. In our case, however, it is not applicable, as the support of the measurements is truncated. Instead, we use a general non-linear curve fitter to fit equation 5.16. In experiments with real data, we found that for most specular materials the peak can reliably be detected with even a single lobe. See figure 5.6 for sample temporal reflectance profiles of measurements of a coin.

In the proposed setup, the two maximum-response peaks μ_h and μ_v of the horizontal and vertical sweep, respectively, directly correspond to a monitor location $(x, y) \propto (\mu_\mathrm{h}, \mu_\mathrm{v})$, which in turn allows us to determine $\boldsymbol{s}_i$.

5.6.2 Properties

In general, the approach requires free lines of sight between the surface point and the two cameras, as well as from the point to the two light sources in the reflection direction. This imposes constraints to the object geometry. Similar constraints are also present with other photometric reconstruction techniques, such as photometric stereo. Unlike photometric stereo, which suffers from normal bias in the case of (partial) self-shadowing, specularity consistency only requires a fairly narrow free cone of sight to the lighting environment and is hence more robust against normal bias by self-shadowing.

Figure 5.5: Photograph of our experimental setup. The object is placed in front of the monitor, facing the camera pair. The cameras capture image pairs while the monitor displays sweeping linear light sources. The small projector visible above the cameras is used for comparison with structured-light reconstructions.

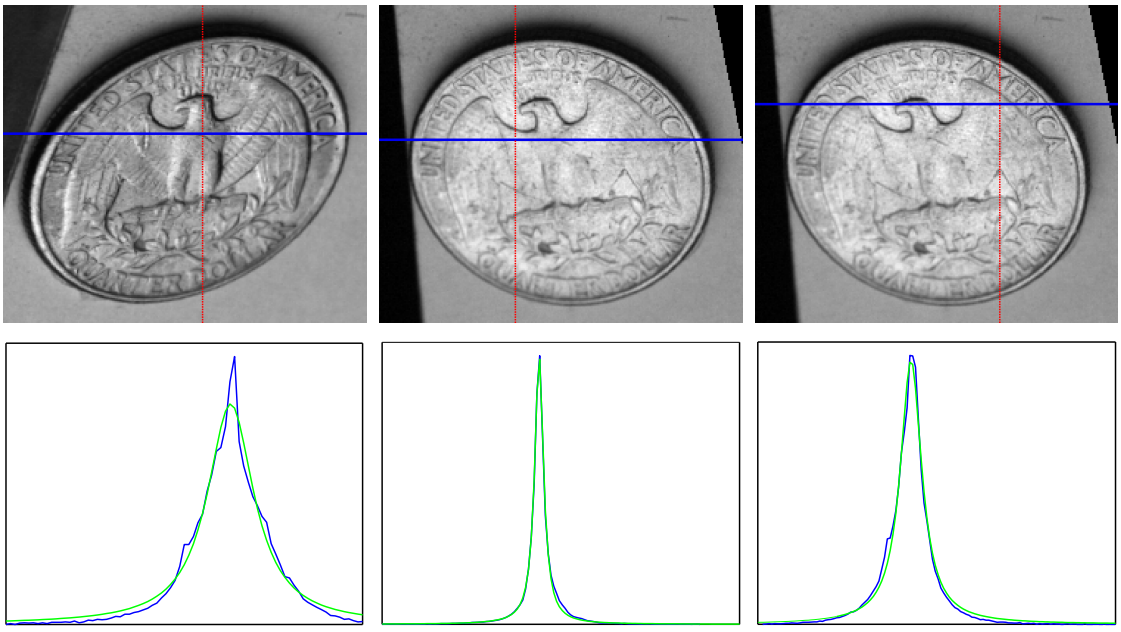


Figure 5.6: (Top) Temporal response measurements at different points of a coin. (Bottom) Corresponding measured intensity profiles. Measurements are shown in blue, and the fit of a single Gaussian is shown in green.

A practical limitation of the proposed setup, however, is the relatively small solid angle covered by the monitor. In order for a point to be reconstructed, the virtual image of (potentially different) areas of the monitor must be visible by both cameras observing the point. Only points with normals falling within a narrow cone satisfy this requirement. Normals falling outside of this cone result in gaps in the reconstruction. We are currently investigating alternative layouts that will attenuate this problem by covering a larger solid angle.

5.7 Results

Using the setup described in the previous section, we captured two real datasets: a quarter and a mirrored sphere. In order to evaluate our method in ideal conditions, we also generated synthetic data for a Greek panel and for an analytic sphere, using the same calibration geometry as the real scanner. The simulated datasets use the same pipeline as for the real data, starting with simulated camera images instead of real images.

In order to demonstrate the versatility of the proposed cost function, we use two different stereo reconstruction algorithms. Figure 5.7 shows a reconstruction of the simulated sphere dataset using Markov Random Fields stereo reconstruction [25]. All other reconstructions were obtained using dynamic programming. Results are shown in figure 5.8.

In general, we have found that our technique consistently produces high-quality normals, but can generate incorrect depth estimates (compare the first and second columns of figure 5.8). Due to the geometry of the capturing setup, an incorrect match can result in a large error in the reconstructed depth, whereas the normal estimate computation tends to be more robust to such errors.

Nevertheless, normal and depth are not independent, and we can use the measured normals to improve the quality of the depth estimates. To this end, we use the optimization method described in chapter 4. Recall the idea is to formulate an energy minimization problem whose solution attempts to reconcile both measurements. Since the method expects to correct relatively small depth variations, we iterate the process to produce larger corrections. The third column of figure 5.8 shows the result of such optimization. Comparing with the original depth estimates, the improvements are clear.

The mirrored sphere shows the entire range of orientations that can be reconstructed by our system. As discussed in section 5.6.2, reconstruction is only possible for points within the overlapping images of the monitor as seen by both cameras. For further illustration, figure 5.8h shows the reflection of the monitor in the sphere as seen from one of the cameras.

For the quarter dataset, we also attempted to obtain geometry with a structured light scanner. The projector is used exclusively for this experiment and is visible in figure 5.5. Figure 5.8d shows the resulting reconstruction. Stripe artifacts show the difficulties arising from scanning specular objects with active triangulation. Attempts to obtain the shape of the mirrored sphere using structure light projection consequently failed completely.

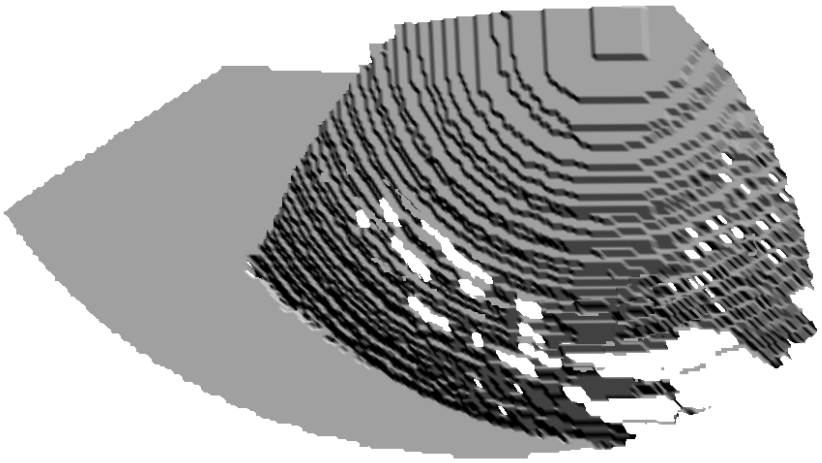
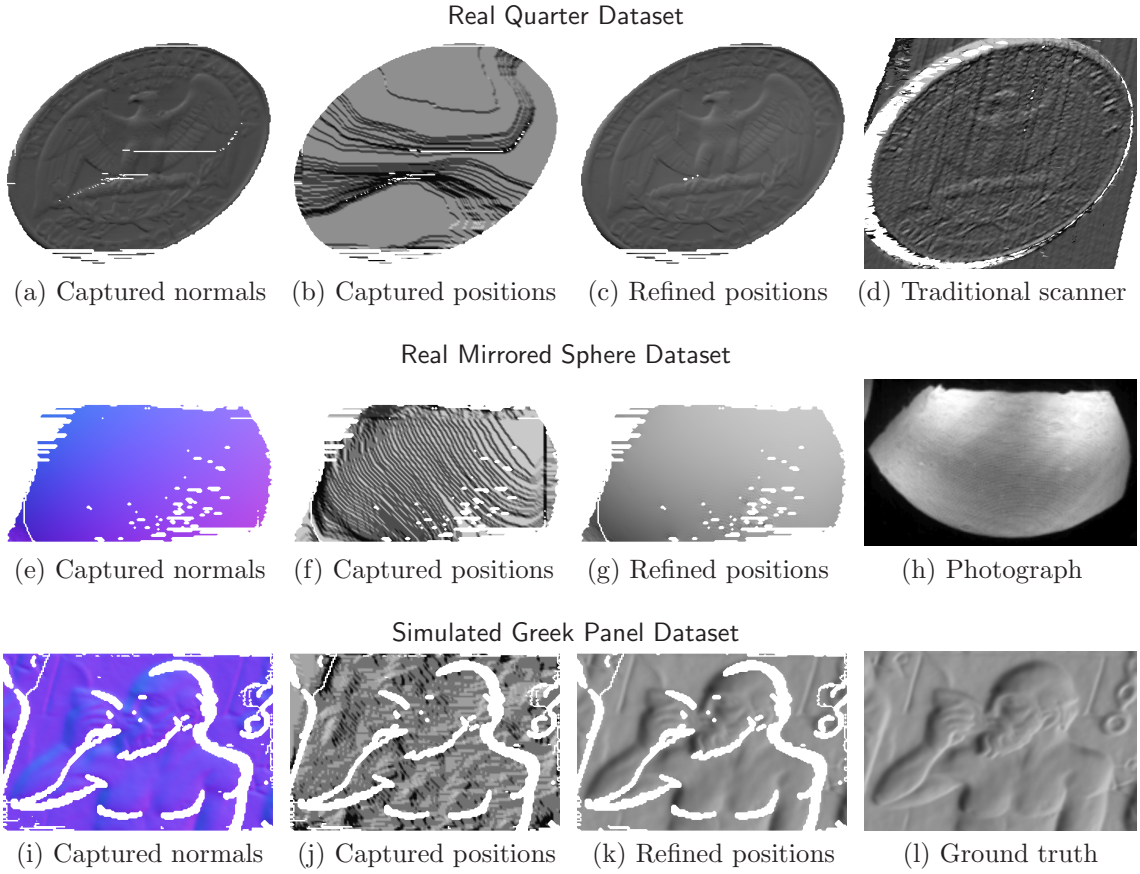


Figure 5.7: Rendering from the discrete matches produced by the Markov Random Fields stereo reconstruction [25], evaluated on a simulated dataset of a sphere using our specularity constraint.



(a) Captured normals (b) Captured positions (c) Refined positions (d) Traditional scanner

(e) Captured normals (f) Captured positions (g) Refined positions (h) Photograph

(i) Captured normals (j) Captured positions (k) Refined positions (l) Ground truth

Figure 5.8: Reconstruction results. Images show the captured normal field, the depth recovered by the stereo matching, and the results of geometry refinement.

5.8 Conclusions

In this chapter, we proposed a novel consistency criterion for dense stereo matching. In contrast to intensity-based stereo matching, it is based on normal estimates from specular reflections. The fact that normals are closely related to the geometry being reconstructed allows us to aggregate matching costs in a meaningful manner, biasing the stereo reconstruction toward the measured surface orientation. In addition, we demonstrate how the normal information can be used to refine the output of the stereo algorithm. We demonstrated the practicability of the proposed criterion using a simple acquisition setup based on two cameras and a monitor. We presented dense stereo reconstructions using specular consistency for both simulated and measured datasets.

Chapter 6

Final remarks

Since the publication of the original conference papers that originated this dissertation's chapters, it has been very gratifying to track the ramifications of our research. For example, Zhu et al. [133] developed an omnidirectional 3D scanner based on the unstructured temporal stereo triangulation method we present in section 2.4. Their system captures the geometry of large environments, such as entire rooms.

The geodesic face scanner dome created by Weyrich et al. [124] originally produced geometry that was unsuitable for their purposes. By using the technique we presented in section 4.4.4, along with a normal map captured by their dome, they were able to greatly improve the acquired geometry. The face models were later used as the basis for the creation of a statistical model for the synthesis of detailed facial geometry [53].

The range-image formulation of section 4.4.3 has also been used in follow-up work. Jones et al. [66] created another geodesic dome, this one capable of scanning dynamic scenes with high-speed cameras. Due to motion, the quality of the geometry returned by their triangulation system was poor. The captured normal fields, on the other hand, were excellent. Using our technique, they were able to produce clean geometry from human actors in motion. Later, Ma et al. [82] introduced a new strategy for capturing normal maps. Using a structured light 3D scanner, their normal field, and our optimization method, they were able to produce extremely high-resolution, high-quality 3D facial surface geometry.

6.1 Future work

In the short term, our work can be improved in many ways. In chapter 2, for example, we suggested the use of a projector to illuminate a scene with random stripe patterns in order to generate the disambiguating information required for triangulation. If a projector pixel covers the area of many camera pixels, there will be ambiguities. It is therefore important for the projector and camera resolutions to be compatible. Unfortunately, off-the-shelf projectors have a much lower resolution than off-the-shelf cameras. The generation of appropriate high resolution projected patterns is an

interesting engineering challenge. Since the patterns do not have to be calibrated, one alternative is to randomly perturb the projector's lens (or even the entire projector assembly) while capturing image pairs. This would shift the projected patterns away from pixel boundaries, thereby effectively increasing their resolution.

Chapter 3 also invites future work. Most sub-pixel estimation methods requires us to reconstruct the sampled matching cost function around a neighborhood of the best integer match. At the same time we published our original work, Psarakis and Evangelidis [100] proposed an alternative approach. Instead of reconstructing the matching cost function, they reconstruct the input images. Operating on continuous images, the matching cost function is also continuous. They report improved sub-pixel matches by cutting this continuous matching cost function in the traditional axis aligned direction. It would be trivial to use their reconstruction strategy with our symmetric sub-pixel refinement algorithm. One wonders if further improvements would result from the combination.

Still in the realm of chapter 3, equation 3.6 suggests another venue for future work. It relates the orientation of the matching ridge with the normal to the surface. Normals are usually computed by combining information from adjacent reconstructed 3D samples. The matching ridge slope, on the other hand, can be inferred from the information needed to produce a single sample. By using three cameras arranged in a triangle, we can in theory compute a normal direction by simply looking at the pairwise matching ridge orientations. It is hard to predict the quality of such a normal field. However, given that the ridge-based normal estimates would be more localized than those computed from reconstructed 3D samples, it is possible the normal field would have a wider frequency content.

Woodham [127] shows how to obtain curvature measurements with the photometric stereo method. Curvatures are derivatives of normals, and are therefore related to second order derivatives of positions. It would be interesting to formulate an optimization problem, similar to what we presented in chapter 4, but that takes into account curvature measurements in addition to orientation measurements. An open question is whether the noise sensitivity of second order derivatives would offset the potential gains introduced by the additional source of information.

It would also be interesting to see commercial scanner manufacturers include normal measurement hardware and software into their products. As we have shown in chapter 4, this hybrid scanner design has the potential to capture high quality geometry, possibly reducing hardware cost.

The specularity triangulation algorithm discussed in chapter 5 could also be extended in a variety of ways. One alternative could be to combine the specularity consistency criterion with a wider variety of consistency metrics, such as passive stereo or spacetime stereo. Another line of research involves the design of scanner layouts that could better explore the specularity constraint. We could, for example, increase the number of monitors to cover a wider cone of normals. Alternatively, we could include more cameras in order to reduce the chances of ambiguities.

Although the acquisition of 3D geometry has become a well established science, there are still many challenges ahead. Certain material categories present significant problems to current

systems. Examples include shiny, extremely dark, transparent, or translucent materials. Objects with high depth complexity are also hard to scan, due to occlusions. In this dissertation, we presented techniques that were robust to individual adversities. However, there are still no practical scanning systems capable of handling *any object*. The design of such a general scanner would be key to digitizing large collections, such as those found in museums. The diversity of materials and shapes comprised in large collections make their capture tedious or even impractical with current technology. The contributions in this dissertation represent small steps towards this goal.

Bibliography

[1] J. Ahmon. The application of short-range 3D laser scanning for archaeological replica production: The Egyptian tomb of Seti I. *The Photogrammetric Record*, 19(106):111–127, 2004.

[2] N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2–3):127–153, 2001.

[3] K. Araki, Y. Sato, and S. Parthasarathy. High speed rangefinder. In *SPIE: Optics, Illumination, and Image Sensing for Machine Vision II*, volume 850, pages 184–188, 1987.

[4] S. Banerjee, P. S. Sastry, and Y. V. Venkatesh. Surface reconstruction from disparate shading: An integration of shape-from-shading and stereopsis. In *International Conference on Pattern Recognition*, volume 1, pages 141–144, 1992.

[5] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition*, 31(7):963–982, 1998.

[6] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002.

[7] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[8] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin. Building a digital model of Michelangelo's Florentine Pietà. *IEEE Computer Graphics and Applications*, 22 (1):59–67, 2002.

[9] P. J. Besl. *Active Optical Range Imaging Sensors*, chapter 1, pages 1–63. Springer-Verlag, 1988.

[10] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[11] D. N. Bhat and S. K. Nayar. Stereo and specular reflection. *International Journal of Computer Vision*, 26(2):91–106, 1998.

[12] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

[13] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. pages 231–236, 1993.

[14] A. Blake. Specular stereo. In *International Joint Conference on Artificial Intelligence*, volume 2, pages 973–976, 1985.

[15] A. Blake and G. Brelstaff. Geometry from specularities. In *IEEE International Conference on Computer Vision*, pages 394–403, 1988.

[16] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[17] W. Boehler, editor. *Comité Internationale de Photogrammétrie Architecturale Working Group 6*, 2002. International Workshop on Scanning for Cultural Heritage Recording.

[18] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, pages 7–56, 1987.

[19] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *IEEE International Conference on Computer Vision*, pages 591–596, 2003.

[20] T. Bonfort, P. Sturm, and P. Gargallo. General specular surface triangulation. In *Proceedings of the Asian Conference on Computer Vision*, volume II, pages 872–881, 2006.

[21] N. Borghese, G. Ferrigno, G. Baroni, A. Pedotti, S. Ferrari, and R. Savare. Autoscan: A flexible and portable 3D scanner. *IEEE Computer Graphics and Applications*, 18(3):38–41, 1998.

[22] J-Y. Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc`, 2004.

[23] J.-Y. Bouguet and P. Perona. 3D photography on your desk. In *IEEE International Conference on Computer Vision*, pages 43–50, 1998.

[24] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):14–28, 1987.

[25] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, 1998.

[26] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[27] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2007)*, 26(3), 2007.

[28] C. Chen, Y. Hung, C. Chiang, and J. Wu. Range data acquisition using color structured lighting and stereo vision. *Image and Vision Computing*, 15(6):445–456, 1997.

[29] C-Y. Chen, R. Klette, and C-F. Chen. Shape from photometric-stereo and contours. In *International Conference on Computer Analysis of Images and Patterns*, pages 377–384, 2003.

[30] T. Chen, M. Goesele, and H-P. Seidel. Mesostructure from specularity. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1825–1832, 2006.

[31] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[32] E. N. Coleman and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. *Computer Graphics and Image Processing*, 18:309–328, 1982.

[33] D. B. Cooper, A. Willis, S. Andrews, J. Baker, Y. Cao, D. Han, K. Kang, W. Kong, F. F. Leymarie, X. Orriols, S. Velipasalar, E. L. Vote, M. S. Joukowsky, B. B. Kimia, D. H. Laidlaw, and D. Mumford. Assembling virtual pots from 3D measurements of their fragments. In *Proceedings of the ACM Conference on Virtual Reality, Archeology, and Cultural Heritage*, pages 241–254, 2001.

[34] J. E. Cryer, P-S. Tsai, and M. Shah. Integration of shape from shading and stereo. *Pattern Recognition*, 28(7):1033–1043, 1995.

[35] B. Curless. *New Methods For Surface Reconstruction From Range Images*. PhD thesis, Stanford University, 1997.

[36] B. Curless. Overview of active vision techniques. In *SIGGRAPH 99 Course on 3D Photography*, 1999.

[37] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *IEEE International Conference on Computer Vision*, pages 987–994, 1995.

[38] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. of ACM SIGGRAPH 1996*, pages 303–312, 1996.

[39] Cyberware Inc. `http://www.cyberware.com`, 2007.

[40] J. Davis and X. Chen. A laser range scanner designed for minimum calibration complexity. In *International Conference on 3-D Digital Imaging and Modeling*, pages 91–98, 2001.

[41] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *CVPR*, volume II, pages 359–366, 2003.

[42] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):296–302, 2005.

[43] T. Davis. CHOLMOD: A set of ANSI C routines for sparse cholesky factorization and update/downdate. `http://www.cise.ufl.edu/research/sparse/cholmod`, 2007.

[44] U. Dhond and J. Aggarwal. Structure from stereo—a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1489–1510, 1989.

[45] R. Epstein, A. L. Yuille, and P. N. Belhumeur. Learning object representations from lighting variations. In *European Conference on Computer Vision*, pages 179–199, 1996.

[46] Eyetronics Inc. `http://www.eyetronics.com`, 2007.

[47] M. Farouk, I. El-Rifai, S. El-Tayar, H. El-Shishiny, M. Hosny, M. El-Rayes, J. Gomes, F. Giordano, H. Rushmeier, F. Bernardini, and K. Magerlein. Scanning and processing 3D objects for web display. In *International Conference on 3-D Digital Imaging and Modeling*, pages 310–317, 2003.

[48] R. Fontana, M. Greco, M. Materazzi, E. Pampaloni, L. Pezzati, C. Rocchini, and R. Scopigno. Three-dimensional modelling of statues: the Minerva of Arezzo. *Journal of Cultural Heritage*, 3(4):325–331, 2002.

[49] R. W. Frischholz and K. P. Spinnler. Class of algorithms for real-time subpixel registration. In *SPIE: Computer Vision for Industry*, volume 1989, pages 50–59, 1993.

[50] P. Fua and Y. G. Leclerc. Using 3-dimensional meshes to combine image-based and geometry-based constraints. In *European Conference on Computer Vision*, volume 2, pages 281–291, 1994.

[51] A. Gardner, C. Tchou, T. Hawkins, and P. Debevec. Linear light source reflectometry. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2003)*, 22(3):749–758, 2003.

[52] A. S. Georghiades. Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In *IEEE International Conference on Computer Vision*, volume 2, pages 816–823, 2003.

[53] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2006)*, 25(3), 2006.

[54] G. Guidi, J.-A. Beraldin, and C. Atzeni. High-accuracy 3D modeling of cultural heritage: the digitizing of Donatello's "Maddalena". *IEEE Transactions on Image Processing*, 13(3): 370–380, 2004.

[55] O. Hall-Holt and S. Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. In *IEEE International Conference on Computer Vision*, pages 359–366, 2001.

[56] M. A. Halstead, B. A. Barsky, S. A. Klein, and R. B. Mandell. Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In *Proc. of ACM SIGGRAPH 1996*, pages 335–342, 1996.

[57] B. K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, Massachusetts Institute of Technology, 1970.

[58] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2006)*, 25(3):569–578, 2006.

[59] K. Ikeuchi. Constructing a depth map from images. A.I. Memo AIM-744, Artificial Intelligence Laboratory, MIT, 1983.

[60] K. Ikeuchi. Determining a depth map using a dual photometric stereo. *International Journal of Robotics Research*, 6(1):15–31, 1987.

[61] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1–3):141–184, 1981.

[62] K. Ikeuchi, A. Nakazawa, K. Hasegawa, and T. Ohishi. The great Buddha project: modeling cultural heritage for VR systems through observation. In *IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 7–16, 2003.

[63] S. Inokuchi, K. Sato, and F. Matsuda. Range-imaging for 3D object recognition. In *International Conference on Pattern Recognition*, pages 806–808, 1984.

[64] J. Jalkio, R. Kim, and S. Case. Three dimensional inspection using multistripe structured light. *Optical Engineering*, 24(6):966–974, 1985.

[65] R. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):122–139, 1983.

[66] A. Jones, A. Gardner, M. Bolas, I. McDowall, and P. Debevec. Simulating spatially varying lighting on a live performance. In *European Conference on Visual Media Production*, pages 127–133, 2006.

[67] T. Kanade, A. Gruss, and L. Carley. A very fast VLSI rangefinder. In *IEEE Transactions on Robotics and Automation*, pages 1322–1329, 1991.

[68] S. Kang, J. Webb, C. Zitnick, and T. Kanade. A multibaseline stereo system with active illumination and real-time image acquisition. In *IEEE International Conference on Computer Vision*, pages 88–93, 1995.

[69] S. Kawada, editor. *International Conference on 3-D Digital Imaging and Modeling*, 2003. Special section on Heritage Applications.

[70] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 61–70, 2006.

[71] D. Koller, J. Trimble, T. Najbjerg, N. Gelfand, and M. Levoy. Fragments of the City: Stanford's Digital Forma Urbis Romae Project. In *Proceedings of the Third Williams Symposium on Classical Architecture*, volume 61, pages 237–252, 2006.

[72] R. Kolluri, J.-R. Shewchuk, and J.-F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 11–21, 2004.

[73] A. Koschan, V. Rodehorst, and K. Spiller. Color stereo vision using hierarchical block matching and active color illumination. In *International Conference on Pattern Recognition*, volume 1, pages 835–839, 1996.

[74] K. N. Kutulakos and E. Steger. A theory of refractive and specular 3D shape by light-path triangulation. In *IEEE International Conference on Computer Vision*, pages 1448–1455, 2005.

[75] H. Lange. Advances in the cooperation of shape from shading and stereo vision. In *International Conference on 3-D Digital Imaging and Modeling*, pages 46–58, 1999.

[76] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H-P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics*, 22(2): 234–257, 2003.

[77] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. In *Proc. of ACM SIGGRAPH 2000*, pages 131–144, 2000.

[78] Y. Li, S. Lin, H. Lu, S. B. Kang, and H-Y. Shum. Multibaseline stereo in the presence of specular reflections. In *International Conference on Pattern Recognition*, volume 3, pages 573–576, 2002.

[79] S. Lin, Y. Li, S. B. Kang, X. Tong, and H-Y. Shum. Diffuse-specular separation and depth recovery from image sequences. In *European Conference on Computer Vision*, volume 3, pages 210–224, 2002.

[80] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 125–131, 1999.

[81] M. I. A. Lourakis. levmar: Levenberg-Marquardt non-linear least squares algorithms in C/C++. `http://www.ics.forth.gr/~lourakis/levmar`, 2004.

[82] W.-C. Ma, T. Hawkins, P. Peers, C.-F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. In *Eurographics Symposium on Rendering*, pages 183–194, 2007.

[83] D. Martin, editor. *IEEE Conference on Computer Vision and Pattern Recognition*, 2003. Workshop on Applications of Computer Vision in Archaeology.

[84] Mathematics and Computer Science Division of the Argonne National Laboratory. PETSC: a portable, extensible toolkit for scientific computation. `http://www.mcs.anl.gov/petsc`, 2004.

[85] G. Medioni and J. Jezouin. An implementation of an active stereo range finder. In *Topical Meeting on Machine Vision*, volume 12 of *Technical Digest Series*, pages 34–51. Optical Society of America, 1987.

[86] G. Miller. Efficient algorithms for local and global accessibility shading. In *Proc. of ACM SIGGRAPH 1994*, pages 319–326, 1994.

[87] M. G. Mostafa, S. M. Yamany, and A. A. Farag. Integrating shape from shading and range data using neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 15–20, 1999.

[88] S. K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431, 1990.

[89] S. K. Nayar, X-S. Fang, and T. Boult. Stereo and specular reflection. *International Journal of Computer Vision*, 21(3):163–186, 1997.

[90] D. Nehab, S. Rusinkiewicz, and J. Davis. Improved sub-pixel stereo correspondences through symmetric refinement. In *IEEE International Conference on Computer Vision*, pages 557–563, 2005.

[91] D. Nehab, S. Rusinkiewicz, J. E. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2005)*, 24(3):536–543, 2005.

[92] Institute of Robotics and German Aerospace Center Mechatronics. The DLR camera calibration toolbox. `http://www.robotic.dlr.de/callab`, 2006.

[93] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):139–154, 1985.

[94] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *International Journal of Computer Vision*, 7(2):143–162, 1992.

[95] M. Oren and S. K. Nayar. A theory of specular surface geometry. *International Journal of Computer Vision*, 24(2):105–124, 1997.

[96] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[97] G. Papaioannou and E. A. Karabassi. On the automatic assemblage of arbitrary broken solid artefacts. *Image and Vision Computing*, 21(5):401–412, 2003.

[98] D. Poussart and D. Laurendeau. *3-D Sensing for Industrial Computer Vision*, chapter 3, pages 122–159. Springer-Verlag, 1988.

[99] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, chapter 10.1. Cambridge University Press, 1992.

[100] E. Z. Psarakis and G. D. Evangelidis. An enhanced correlation-based method for stereo correspondence with sub-pixel accuracy. In *IEEE International Conference on Computer Vision*, volume 1, pages 907–912, 2005.

[101] K. Pulli. Multiview registration for large datasets. In *International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.

[102] K. Pulli, H. Abi-Rached, T. Duchamp, L. Shapiro, and W. Stuetzle. Acquisition and visualization of colored 3D objects. In *International Conference on Pattern Recognition*, pages 11–15, 1998.

[103] T. Roesgen. Optimal subpixel interpolation in particle image velocimetry. *Experiments in Fluids*, 35:252–256, 2003.

[104] H. Rushmeier and F. Bernardini. Computing consistent normals and colors from photometric data. In *International Conference on 3-D Digital Imaging and Modeling*, pages 99–108, 1999.

[105] H. Rushmeier, J. Gomes, F. Giordano, H. El-Shishiny, K. Magerlein, and F. Bernardini. Design and use of an in-museum system for artifact capture. In *CVPR Workshop on Applications of Computer Vision in Archaeology*, volume 1, 2003.

[106] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2002)*, 21(3):438–446, 2002.

[107] P. Saint-Marc, J. Jezouin, and G. Medioni. A versatile PC-based range finding system. *IEEE Transactions on Robotics and Automation*, 7(2):250–256, 1991.

[108] A. C. Sanderson, L. E. Weiss, and S. K. Nayar. Structured highlight inspection of specular surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(1):44–55, 1988.

[109] S. Savarese, M. Chen, and P. Perona. Local shape from mirror reflections. *International Journal of Computer Vision*, 64(1):31–67, 2005.

[110] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, 2002.

[111] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–202, 2003.

[112] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, 1998.

[113] Y. Y. Schechner, S. K. Nayar, and P. N. Belhumeur. A theory of multiplexed illumination. In *IEEE International Conference on Computer Vision*, volume 2, pages 808–815, 2003.

[114] S. A. Shafer. Using color to separate reflection components. *COLOR research and applications*, 10(4):210–218, 1985.

[115] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. *Lecture Notes in Computer Science*, 2350:753–768, 2002. Proceedings of the European Conference on Computer Vision Part I.

[116] M. Shimizu and M. Okutomi. Precise sub-pixel estimation on area-based matching. In *IEEE International Conference on Computer Vision*, pages 90–97, 2001.

[117] J. E. Solem, H. Aanaes, and A. Heyden. A variational analysis of shape from specularities using sparse data. In *3D Data Processing Visualization And Transmission*, pages 26–33, 2004.

[118] T. C. Strand. Optical three-dimensional sensing for machine vision. *Optical Engineering*, 24 (1):33–40, 1985.

[119] R. Szeliski and D. Scharstein. Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):419–425, 2004.

[120] H. D. Tagare and R. J. P. deFigueiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):133–152, 1991.

[121] M. Tarini, H. P. A. Lensch, M. Goesele, and H-P. Seidel. 3D acquisition of mirroring objects. Technical Report MPI-I-2003-4-001, Max-Planck-Institut für Informatik, 2003.

[122] J. Taylor, J.-A. Beraldin, G. Godin, L. Cournoyer, R. Baribeau, F. Blais, M. Rioux, and J. Domey. NRC 3D imaging technology for museum and heritage applications. *The Journal of Visualization and Computer Animation*, 14(3):121–138, 2003.

[123] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):417–438, 1988.

[124] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H. W. Jensen, and M. Gross. Analysis of human faces using a measurement-based skin reflectance model. pages 1013–1024, 2006.

[125] L. B. Wolff and T. E. Boult. Constraining object features using a polarization reflectance model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):635–657, 1991.

[126] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[127] R. J. Woodham. Determining surface curvature with photometric stereo. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 36–42, 1989.

[128] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 274–279, 1993.

[129] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *3D Data Processing Visualization And Transmission*, 2002.

[130] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 367—374, 2003.

[131] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: High-resolution capture for modeling and animation. In *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)*, pages 548–558, 2004.

[132] J. Y. Zheng and A. Murata. Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):913–920, 2000.

[133] J. Zhu, G. Humphreys, D. Koller, S. Steuart, and R. Wang. Fast omni-directional 3D scene acquisition with an array of stereo cameras. In *International Conference on 3-D Digital Imaging and Modeling*, 2007. (to appear).

[134] T. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2/3):215–227, 2002.

[135] P. Zisserman, A. Giblin and A. Blake. The information available to a moving observer from specularities. *Image and Vision Computing*, 7(1):38–42, 1989.

[136] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *Proc. of ACM SIGGRAPH 1999*, pages 205–214, 1999.