LINEAR, DISCRETE, AND QUADRATIC CONSTRAINTS IN SINGLE-IMAGE 3D RECONSTRUCTION

by

Ady Ecker

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Linear, Discrete, and Quadratic Constraints in Single-Image 3D Reconstruction

Ady Ecker

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2010

In this thesis, we investigate the formulation, optimization and ambiguities in single-image 3D surface reconstruction from geometric and photometric constraints. We examine linear, discrete and quadratic constraints for shape from planar curves, shape from texture, and shape from shading.

The problem of recovering 3D shape from the projection of planar curves on a surface is strongly motivated by perception studies. Applications include single-view modeling and uncalibrated structured light. When the curves intersect, the problem leads to a linear system for which a direct least-squares method is sensitive to noise. We derive a more stable solution and show examples where the same method produces plausible surfaces from the projection of parallel (non-intersecting) planar cross sections.

The problem of reconstructing a smooth surface under constraints that have discrete ambiguities arise in areas such as shape from texture, shape from shading, photometric stereo and shape from defocus. While the problem is computationally hard, heuristics based on semidefinite programming may reveal the shape of the surface.

Finally, we examine the shape from shading problem without boundary conditions as a polynomial system. This formulation allows, in generic cases, a complete solution for ideal polyhedral objects. For the general case we propose a semidefinite programming relaxation procedure, and an exact line search iterative procedure with a new smoothness term that favors folds at edges. We use this numerical technique to inspect shading ambiguities.

Acknowledgements

I would like to express my deepest gratitude to my supervisors Kyros Kutulakos and Allan Jepson. It was Kyros who introduced me to the world of 3D vision, and Allan who introduced me to the world of numerical computation. This thesis would have been impossible without their guidance and support.

I wish to thank my committee members Sven Dickinson and David Fleet for making our vision group as active and diverse as it is, and Jianbo Shi for serving as the external examiner.

Many thanks are due to all my officemates at U of T for enriching my learning experience, and especially Sam Hasinoff, Faisal Zubair-Qureshi, Stratis Ioannidis, Jingrui Zhang, Michael Daum, Martin de Lasa, Jacobo Bibliowicz and Peter O'Donovan. John Hancock deserves credit for helping with many of our lab experiments.

I would like to thank the former and present members of our vision group, including Gustavo Carneiro, Francisco Estrada, Diego Macrini, Nigel Morris, Pablo Sala, Michael Jamieson, Alex Levinshtein, Fernando Flores-Mangas, Marcus Brubaker and Jonathan Taylor.

Finally, I am grateful to my family for their support, and to Carol Jepson for her donation.

“In the past, research in the pattern recognition field has been limited to the identification of two-dimensional shapes, mainly because it was thought that any three-dimensional analysis would be more difficult. The idea seems to have been that the 2-D work would pave the road for future 3-D work. However, progress has been slow and it may well be that the study of three-dimensional projections is an easier step.”

Lawrence G. Roberts, 1963

Contents

Chapter 1

Introduction

1.1 Overview

The single-image 3D reconstruction problem is to recover a 3D surface from its 2D image. The motivation to study this inverse problem is that 3D shape is an inherent invariant of objects to variations in pose, illumination conditions and surface markings. This invariance can be very useful for recognition and categorization. In addition, 3D information makes the problem of segmenting a scene into different bodies much simpler. While most current methods for segmentation and recognition are based on 2D appearance models, they do not generalize far beyond their training set. It is almost impossible for 2D-based methods to identify two objects with the same 3D shape but very different texture or surface markings as similar objects. 2D appearance-based systems are also limited in their ability to interpret articulated objects and occlusion [130]. Therefore, both 2D and 3D processing are essential to achieve generic recognition and categorization. Although technologies such as laser range finders and multiple view geometry can provide accurate surface reconstructions, the single-image problem is of fundamental theoretical and practical importance to the science of vision.

This thesis examines the formulation, optimization and ambiguities in single-image 3D surface reconstruction from geometric and photometric constraints. The thesis consists of in-

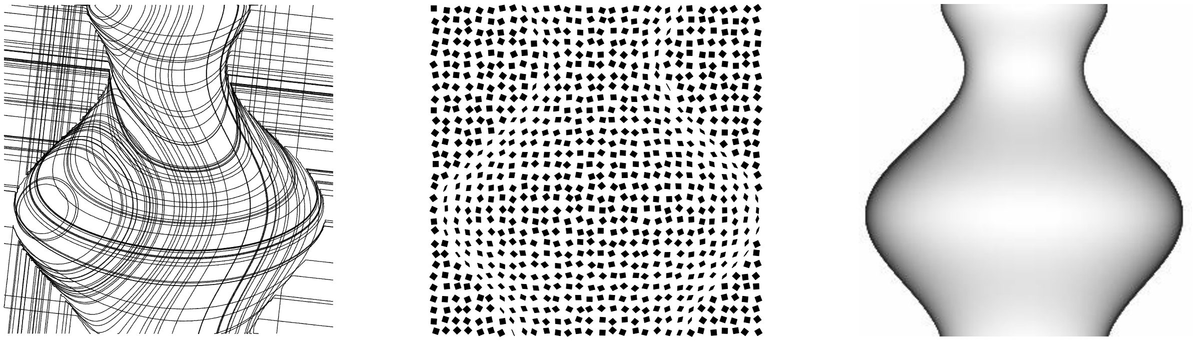


Figure 1.1: The three types of single-image 3D reconstruction problems we will deal with are shape from planar curves (left), shape from texture (middle), and shape from shading (right). In each case, the goal is to come up with a plausible 3D interpretation of the scene (or a family of 3D interpretations). Although these specific images are different at the local level, humans can identify their three-dimensional shape similarity.

troductory and research chapters. Chapters 1, 2 and 5 contain background material. Rather than being comprehensive surveys, they are meant to provide a high-level conceptual discussion of previous approaches, highlight points that are less widely known, and direct the reader to the relevant literature. Chapters 3, 4, and 6 are expanded versions of our papers [39–41]. They present our research on shape from planar curves, shape from texture and shape from shading. Examples of the sort of images that will be examined are shown in figure 1.1. The original contributions in these chapters are quite technical, hence their summary is deferred to chapter 7.

Whereas most work in computer vision is organized by problems and the emphasis is on implementation, the general organization here revolves around mathematical techniques. Instead of presenting a particular system for a particular problem, the thesis identifies similar mathematical structures in seemingly different problems that were treated independently in the past. Compared to the “problem oriented approach”, the “technique oriented approach” is also easier to study and implement with software components. As a consequence, we present more general, robust and cleaner algorithms to some of these problems.

1.2 Reconstruction by humans

Much of computer vision research, including 3D reconstruction, was inspired by human perception. Ideally, we would like computer systems to be able to perform every visual task that humans can do. Theoretical insights about human perception may lead to further generalizations. For example, binocular stereo in computer vision was inspired by human perception, and generalized to multi-camera stereo systems. Although we will not deal directly with humans, the questions of what humans can do and how are always in the background.

There is a lot of evidence suggesting that humans attempt to interpret images as three-dimensional surfaces [75, 106, 112, 129, 134, 142, 171, 193]. The surfaces could be unfamiliar, and their shape never seen before. Even more, the images could be of unnatural images, such as line drawings, textured surfaces, and artificially shaded images, as shown in figure 1.1. While human 3D perception may not be veridical, it is often better than current computer vision systems, especially in real world scenes. All this may suggest that 3D perception is central to human vision, possibly for its role in recognition.

Humans' object recognition had been studied extensively, and a complete account is out of the scope of this thesis. An up-to-date survey is presented in [61]. One key question that had been examined is whether humans perform recognition in a viewpoint-invariant manner. This issue has not been completely settled yet, as both viewpoint-invariant and viewpoint-dependent neurons have been detected [61]. However, even if object recognition by humans is viewpoint-dependent, that does not mean that 3D reconstruction processes are not involved. Due to the inherent ambiguities in single-image 3D reconstruction, it is possible that the estimations of the same surface seen from different viewpoints will not be identical.

1.3 Reconstruction by computers

The need for 3D reconstruction was clear to the pioneers of computer vision, like Roberts [157], Horn [75] and Marr [129]. However, over the years the problem proved to be very difficult.

The primary source of difficulty is the fact that depth information is discarded during projection onto the image plane. That makes the problem ill-posed, since different 3D surfaces can have exactly the same image. In fact, for any pair of images, there exists a Lambertian surface that produces the two images under two illumination conditions (the surface can have arbitrary albedo at every point, and shadows and inter-reflections are ignored) [27]. This implies that no non-trivial function of the image of a surface can be completely invariant to illumination changes. Hence, the problem is severely under-constrained and additional assumptions have to be made.

Fortunately, surfaces in our world have many regular properties that leave a trace in the projected image. Single-image reconstruction methods take advantage of such regularities. They usually deal with the following elements:

1. **Assumptions about the world and image formation process.** Assumptions reflect prior knowledge about surfaces that we expect to see. They may also involve assumptions about the illumination and the way the regularity projects onto the image. The goal of a reconstruction method is to invert the image formation process. Note that the assumption may hold or not, so ideally there should be a way to do a consistency check.

2. **Cue extraction from the image.** This is a low-level image processing step, where elements such as edges and texture features are identified. Perceptual organization and grouping principles might be applied to generate hypotheses about the 3D relations between image features.

3. **Constraints.** Constraints are mathematical conditions that the surface and problem parameters should satisfy. The constraints are derived from the image cues under the assumed image formation model. The goal is to solve for the surface, or family of surfaces, that satisfies the constraints.

4. **Optimization.** Even if the system of constraints fully characterizes the surface we may not be able to find the solution. Optimization complexity is a serious issue, since systems

of nonlinear constraints could be instances of NP-hard problems. The systems tend to be large since they depend on the desired surface resolution. In practice these systems are noisy, and exact solutions may not exist. Therefore, we may need to settle with approximation algorithms that have no quality guarantees.

5. **Ambiguities.** Depth cues are often very ambiguous at the local level. Typically, it is impossible to recover the 3D shape from small image windows, and hence there is a need to integrate the cues over substantial parts of the image. It may also be the case that some global ambiguities remain, for example an in/out Necker reversal. Ambiguities are important to study, since they reveal the limitations of bottom-up reconstruction under a particular set of assumptions, no matter how good the optimization procedure being used is. In order to determine the shape, additional prior knowledge should be used to pick a shape inside the space of ambiguous shapes.

It is evident that the problem is difficult, involving the making of assumptions, large optimization problems, and dealing with ambiguous solutions. For these reasons, after the first wave of pioneering work in computer vision, interest in single image reconstruction declined during the 90s and the problem almost disappeared from computer vision courses and textbooks. Recognition methods turned into appearance and statistical models, while surface reconstruction methods turned towards multiple view geometry. Of course, multiple view methods are not a complete solution to the surface reconstruction problem, the same way that attaching a barcode to objects is not a solution to visual recognition.

In recent years research interest in single image reconstruction has somewhat increased. Apart from the classical computer vision setting in robotics, several new applications are emerging. First, there is now a large amount of unannotated images on the Internet which are of interest to search engines. Even a rough 3D reconstruction, such as identification of buildings and roads, could be a first step towards recovering some information about their content and the person who took them. Second, computer graphics applications increased the demand for 3D models of characters and environments. Interfaces of freeform sketching systems [32, 139] as well as single view modeling from photographs [217] would improve if part of the reconstruction process is automated. These programs range between computer-aided user interfaces and human-assisted computer vision.

1.4 Reconstruction by recognition

The reconstruction paradigm presented thus far is completely bottom-to-top, going from image features to a system of constraints and its solution surface. In this thesis we will focus on the formulation and optimization phases only. To complete the picture we briefly examine here the question whether a top-to-bottom approach can do surface reconstructions without going through the complicated optimization phase.

Roberts' thesis [157] was the first attempt in computer vision to fit known 3D models of polyhedra to an image. After preprocessing the image with an edge-detector and detecting corners, the program solved for 3D poses of the polyhedral models so that the projection of their 3D vertices onto the image would be close to the image corners. Rather than using 3D reconstruction to aid recognition, in this approach 3D reconstruction is essentially a byproduct of the recognition process. The approach is limited since except for very restricted domains (e.g. industrial parts), we cannot assume a-priori knowledge of the surfaces we might encounter.

One way to increase the amount of shapes that can be resolved is to fit a linear combination of shape models. For instance, Atick [3] demonstrated the use of shapes basis for approximate 3D reconstruction of faces from single images. We will discuss in detail linear subspaces derived from planar curves in chapters 2 and 3. This approach is appealing, since once the category of an object is identified the search for the shape can be carried out in a low dimensional subspace. However, real-world shapes can be quite complicated and generally cannot be approximated by low-dimensional linear subspaces. This is in contrast to diffuse BRDFs and global illumination for which low-dimensional models make reasonable approximations. The problem is that as the image resolution is scaled up, the possible variability of shapes increases beyond what low-dimensional models of the entire image can represent.

To get beyond the limitations of entire surface recognition it was suggested to perform 3D reconstruction by recognizing image parts. The idea of recognizing parts is old (e.g. Marr [129]). However, some of the more recent methods use learned surface patches instead of volumetric primitives. In its radical version [51, 67, 73, 78, 135, 168], the only 3D-aware component is a recognition module that proposes candidate surface parts for each image part. This leads to a labeling problem, where the goal is to choose among the proposed surface parts those that are compatible with their neighbors. Probabilistic formulations of the problem use Markov Random Fields (MRF), similar to other labeling problems and texture synthesis in particular.

Reconstruction by recognition of parts was demonstrated on some real images that were impossible to reconstruct with previous edge-based and shape from shading methods [168]. Yet, there are limits to the generality of this approach. On the one hand, small image patches can be very ambiguous locally. For instance, it is impossible to estimate the orientation (and depth) of a constant-intensity patch. In fact, as we shall see in chapter 6, its shape need not be planar. This is reminiscent of the problems in Pentland's local shape from shading approach [75]. On the other hand, large image patches require an image-shape dictionary whose size grows exponentially with the patch size. Furthermore, the resulting labeling problems are generally NP-hard.

A more traditional approach combines learned statistics with geometric and photometric constraints. Partial recognition can be used to establish the system of constraints or resolve surface ambiguities. For example, Han and Zhu [66] learned 3D edges of cloth and used these edges as boundary conditions for shape from shading. Lipson and Shpitalni [120] used statistics of angles to improve the interpretation of ambiguous polyhedra. The sketching system of Chen et al. [28] used a database of detailed 3D models in conjunction with a traditional analysis of edges and corners.

In conclusion, although we will not deal with recognition in this thesis, there is definitely room for additional learned knowledge for setting constraints on the shape. Note, however, that while recognition can be done by detecting some distinctive features, the nature of the 3D reconstruction problem is very different, requiring the assignment of depth at every surface point.

1.5 Constraints formulation

In this section we give a brief introduction to the formulation of constraints on surfaces. Here and throughout most of the thesis we assume an orthographic projection. Before we can specify constraints we need to choose a representation for the surface. The common practice is to represent the surface as a function z(x,y), or another linear surface representation (spline). Other surface representations, notably level sets and point clouds, could be more appropriate for a moving surface with topology changes. Since we deal with a single image these representations have no advantage.

In a linear surface representation a set of N points (x_i, y_i, z_i) is related to a vector of parameters $\mathbf{v}$ via basis functions $(z_1, \ldots, z_N)^T = \mathbf{B}\mathbf{v}$, where we assume that (x_i, y_i) are known, and z_i are the unknowns.

Throughout the thesis we will meet several such linear representations. For example, a height map on a grid has the trivial basis $\mathbf{B} = \mathbf{I}$, $\mathbf{v} = (z_1, \ldots, z_N)^T$. In chapter 4 we will work with a low-dimensional spline basis made of smooth functions. In Chapter 6 we will look at triangular meshes. In choosing a basis there is a tradeoff between the number of basis functions, which affects the computational cost, and the fidelity of the representation. However, mathematically all linear representations are similar in the sense that one can perform an approximate conversion between any two linear representations. That is, if $(z_1, \ldots, z_N)^T \approx \mathbf{B}_1\mathbf{v}_1 \approx \mathbf{B}_2\mathbf{v}_2$, we can use the pseudo-inverse operator to obtain $\mathbf{v}_1 \approx \mathbf{B}_1^+\mathbf{B}_2\mathbf{v}_2$. Thus, constraints specified in one linear representation can be cast to any other linear representation. Obviously, the loss of accuracy in such conversions depends on the fidelity of the bases.

The advantage of linear representations is that surface depth and derivatives are simple

Table 1.1: Common linear constraints.

Constraint	Formulation	Possible sources
known depth	$z_i = z_0$	stereo or symmetry
co-planar points	$z_i = ax_i + by_i + d$	(see chapter 2)
known normal	$p_i = p_0$, $q_i = q_0$	brightest point in the Lambertian model, vanishing line of a plane in perspective
known normal tilt	$q_i = p_i \tan \tau$	occluding contour and shape from texture

linear functions. For example, on a grid we may approximate the surface derivatives using finite differences $\mathbf{p}_{ij} = \mathbf{z}_{i+1,j} - \mathbf{z}_{ij}$, $\mathbf{q}_{ij} = \mathbf{z}_{i,j+1} - \mathbf{z}_{ij}$. The surface normal is a vector in direction $(-p, -q, 1)$. It is sometimes convenient to represent a unit normal vector with slant σ and tilt τ angles. The (slant,tilt) representation is similar to (latitude,longitude) in geography. In such a representation a unit normal vector is $(\sin\sigma\cos\tau, \sin\sigma\sin\tau, \cos\sigma)$.

Next we turn to discuss various types of constraints. Surprisingly, most single-image constraints studied in computer vision fall into only three categories.

1. **Linear constraints**. Linear constraints are by far the most studied class of constraints. Some examples are given in table 1.1. We will examine co-planarity constraints in detail in chapters 2 and 3. Note that linear constraints may be specified with equalities or inequalities.

2. **Discrete constraints**. We call a discrete constraint a case where some known linear function of the surface can attain finitely many known values. The most common case is binary (sign) ambiguity, due to in/out Necker reversal. This ambiguity appears in the analysis of shape from texture that we will examine in detail in chapter 4. An important binary ambiguity is Perkins' cubic corner [84, 198]. Interestingly, the perspective shape-from-angle problem can have more than two solutions [210, 212].

3. **Quadratic constraints**. Quadratic constraints show up naturally in problems that involve known angles. For example, an orthogonality constraint between two planar faces with normals $(-p_1, -q_1, 1)$ and $(-p_2, -q_2, 1)$ can be written as $p_1q_1 + p_2q_2 + 1 = 0$. In chapter 6 we will discuss quadratic constraints in shape from shading.

Obviously, there are constraints that don't fit nicely into the classification above. Notable exceptions are non-Lambertian reflectance functions (e.g. Phong's model). However, if specular reflection can be separated from diffuse reflectance and the illumination direction is known, there is actually a linear constraint at specular points (the normal is the bisector of the viewing and illumination directions).

1.6 Optimization

The formulation of constraints leads to an optimization problem, whose difficulty depends on the type of constraints being applied. Systems of linear constraints are tractable, as long as they are well conditioned. Linear inequality constraints lead to linear programming (LP). Linear equality constraints can be solved by procedures of numerical linear algebra. For the size of problems we work with in chapter 2, the SVD procedure is generally accurate. Note that there is extensive literature on iterative methods for large sparse linear systems. However, when it comes to the computation of null vectors, which are essential to obtain the space of solutions, iterative methods [60, 99] are less reliable, or might be impractically slow for very large systems.

Unlike linear constraints, discrete ambiguities lead to combinatorial problems. For small number of discrete constraints an exhaustive search might still be feasible, but many of these problems are NP-hard. Continuous quadratically constrained quadratic programs (QCQP) are generally NP-hard as well. Note that QCQP subsumes both LP and problems with discrete binary ambiguities.

In chapters 4 and 6 we will apply semidefinite programming (SDP) relaxations to discrete ambiguities in surface reconstruction and quadratic constraints in shape from shading. SDP is currently a very active research area with a wide range of applications. For a general introduction to SDP see [68, 115, 189]. Recently, SDP-based approximation algorithms have been developed for several computer vision problems, such as image restoration [94, 95], segmentation [95, 140], graph matching [4, 169, 215] and finding correspondences in stereo [190].

While SDP methods are slower than many other iterative methods, they have several attractive properties. First, for some combinatorial problems, notably max-cut [59], SDP-based algorithms provide the best currently known approximation ratio. Second, SDP methods do not depend on an initial guess. Third, SDP solvers can integrate linear constraints, which are problematic for other nonlinear optimization methods. Fourth, tight bounds from convex relaxations can be used in branch-and-bound schemes to find globally optimal solutions [23, 24, 155]. Last, SDP has inspired faster iterative algorithms of similar quality in practice but without any guarantee [18]. Thus, SDP relaxations are radically different from shape optimization methods used in the past in computer vision.

Chapter 2

Planarity

2.1 History of planarity constraints

The problem of remotely measuring distances, and the perception of surface shape, have occupied artists, opticians, photogrammeters, psychologists and many others long before the age of computers. In computer vision, Roberts' thesis [157] from 1963 marks the beginning of investigations into single-image 3D reconstruction. It was only natural for the early works to start by examining polyhedral objects. The reason is that intersecting planar faces of polyhedra generate straight edges (brightness discontinuities) in the image, which are relatively easy to detect. Essentially, the problem reduces to making a 3D interpretation out of an edge map or a line drawing. Despite the simplicity of planar faces their theory is not trivial. In this chapter we survey the historical development, motivations, and formulations of co-planarity constraints. Additional historical accounts can be found in [158, 183, 197] and the first edition of [37].

Roberts' thesis dealt with objects whose shape was known beforehand. The next logical step, carried out by Guzman [64], was to try to identify different bodies in an image of unknown objects. Guzman defined types of vertices, such as "fork" or "T-junction", and realized that certain types cannot be linked by an edge if the edges were to belong to a single 3D body. He also tested his program on optical illusions such as the Reutersward-Penrose triangle. Tests on impossible objects are important, since although we can generally assume that pictures come from real objects, it is still a good idea for a method to be able to check the internal consistency of its assumptions.

Guzman's program used a collection of heuristics that worked most of the time. In a subsequent work, Huffman [79] and Clowes [30] independently analyzed more rigorously the possible configurations of trihedral vertices (vertices formed by the intersection of three distinct planar surfaces). In their scheme, known as "Huffman-Clowes labeling", each edge is labeled as "convex", "concave" or "occluding". Their junction catalogue contains the labels of edges meeting at a valid trihedral junction. The problem of determining whether a line drawing has no 3D realization became to assign each edge a label such that the labels of edges meeting at a vertex form a valid junction in the restricted catalogue. This labeling problem is NP-hard [97]. However, Waltz [202] showed that after assigning labels around a vertex, the number of possibilities for neighboring vertices reduce drastically. The median-case of labeling a random trihedral drawing is linear in the number of junctions [143].

It is interesting to note that the people involved in these early projects made various contributions outside computer vision. Roberts worked on packet switching and is a pioneer of the internet. Huffman is known for his work on compression. Waltz's filtering algorithm, known as arc-consistency, is an important procedure in the field of constraints satisfaction problems (CSP). Marvin Minsky supervised Guzman and was involved in Waltz's work. He also supervised Horn's pioneering thesis on shape from shading, and is known for his works on artificial intelligence.

Research on line labeling continued in several directions. Huffman [79] and Malik [126] extended the labeling framework to curved surfaces. Varley [197] extended the junction catalogue to tetrahedral vertices. Additional results are covered in a recent book by Cooper [33]. Other works looked at the question of identifying faces from images of polyhedra or wireframes [122, 174, 191].

Even if we ignore the issues of extracting line drawings from real images and the complexity of finding legal labelings, there are several problems with the line labeling approach. First, line labeling does not output 3D surfaces. Second, it was found that the average number of legal labelings of tetrahedral vertices is exponential in the number of vertices [197, 199]. Thus, labelings are insufficient for characterizing 3D shapes. Third, a legal labeling is a necessary but insufficient condition for 3D realizability, as there are illusory drawings that have legal labelings but cannot be realized as 3D objects (note that the NP-hardness result for line labeling [97] does not necessarily reduce SAT formulas to drawings of polyhedra that can be realized as 3D objects).

These problems led Sugihara [183] to add algebraic co-planarity constraints. His conditions are necessary and sufficient for a labeled line drawing to be realizable as a 3D polyhedron. Linear systems of co-planarity constraints already appeared in Falk [42], but Falk did not solve them numerically. When Sugihara attempted to solve these systems, he encountered some numerical issues which will be the subject of chapter 3. Although Sugihara's original method begins with a line labeling, it was later argued that the line labeling step is unnecessary if the algebraic system can be formed [71, 158, 199].

In parallel, and completely independently from the work done on the analysis of polyhedra, there has been work on constraints from planar cross sections. Objects with planar cross sections are very common. They include surfaces of revolution (SOR [31]), and more generally, Straight Homogeneous Generalized Cylinders (SHGC [196]). In addition, linear systems arising from structured light have similar structure [14]. Figure 2.1 illustrates fully uncalibrated structured light. The curves in the image are projections of planar cross sections of the surface. The goal is to compute the 3D surface from the observed network of curves without any information about the planes. The surface at the bottom of figure 2.1 was computed using our method, which will be described in chapter 3. Notice that all planar curves could be placed on the same plane. A key here is to escape from flat and nearly-flat solutions.

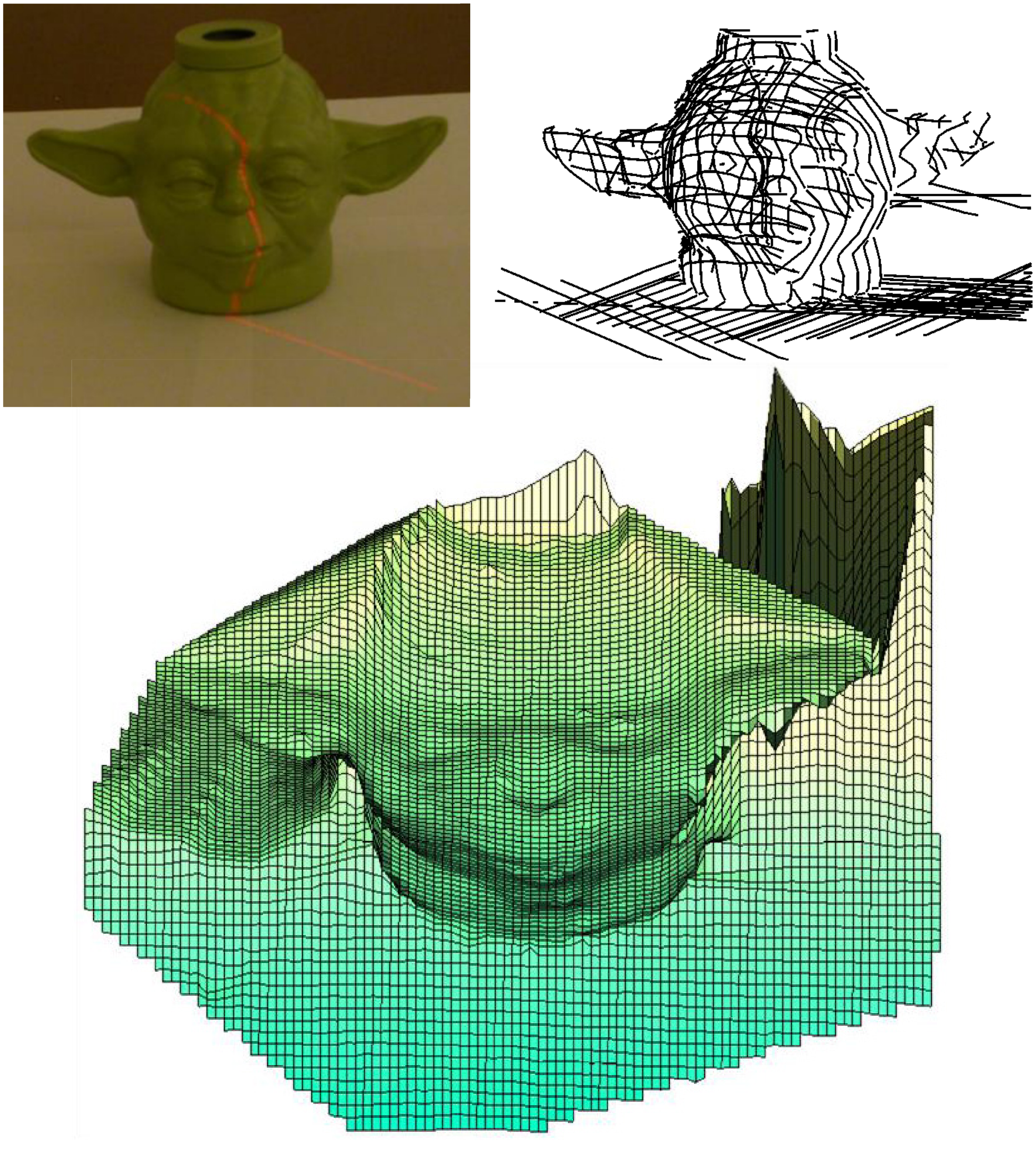


Figure 2.1: 3D reconstruction from uncalibrated structured light. Top left: a planar laser strip projected on the object. Top right: curves extracted from a video sequence of projected strips. Bottom: a rendering of the computed surface.

2.2 Motivations for planarity constraints

There are many reasons to study planarity. Perception studies (e.g. [181, 188]) have demonstrated cases where humans clearly make the planarity assumption. For example, Pizlo et al. [148] showed that planarity can even override stereo depth cues. Although the extraction of salient curves from real images is a difficult problem by itself, it is hard to believe the sophisticated ability to interpret curves as 3D surfaces had been evolved with no relation to perception of real scenes. Why are humans making the planarity assumption? Although there is currently no definitive answer to this question, there are several possible explanations.

A somewhat naive argument for the importance of planarity is that planar curves are common in the environment. A closer look reveals that planar curves are ubiquitous mainly in man-made objects, like buildings. Planarity is justified in artificial environments, but uncommon in nature, albeit several exceptions. In some regions the ground is flat, and the same holds for calm water. Cylindrical and ellipsoidal objects have a planar rim from any viewpoint [100]. Planarity is also important for motion analysis, since objects thrown in the air trace a roughly planar trajectory.

It is worth pointing out that co-planarity subsumes parallelism and co-linearity. There is no need to treat these conditions separately as many sources do. A common principle of perceptual organization is that parallel lines in the image are likely to be the projection of parallel lines in 3D (under orthographic projection). A similar principle is that parallel lines in the image are likely to be the projection of *co-planar* lines in 3D. The subtle difference between the two is that nearly-parallel lines in the image can still be the projection of co-planar 3D lines, which is a linear constraint, in contrast to nonlinear measurements for nearly-parallel lines in 3D (e.g. [119]). The same holds for co-linearity. Points which are nearly-collinear in the image can be assumed co-planar in 3D. We emphasize this point since linear and parallel lines are more common in nature than planar surfaces.

Rothwell et al. [161–163] made an interesting suggestion why an internal representation based on planes might be useful for object recognition. They proposed the “polyhedral cage” metaphor, where a 3D object is surrounded by a virtual polyhedron. The vertices of this polyhedron need not be attached to the surface. The only important requirement is that these vertices can be computed from images of the surface. Once the vertices are identified in an image, we can compute invariants and query a database (geometric hashing [111]). Unfortunately, a theorem of Burns et al. [19] states that there does not exist a function that is view-invariant for all possible point sets of any particular size. This theorem holds for perspective and orthographic projections. However, the fact that there are no invariants for general point sets does not imply that invariants do not exist for special point configurations. The most studied condition where view-invariants exist is co-planarity [154, 162]. Invariants can be written for four co-planar points under orthographic projection and five co-planar points under perspective projection. Note that planarity constraints become meaningful only for four or more points, since three points are trivially co-planar. To be valuable, a polyhedral cage should have some faces with at least four vertices. On such faces affine invariants can be defined. Perspective invariants can be defined on the butterfly configuration of six points on two intersecting planes [162].

2.3 Formulations of planarity constraints

In this section we review the algebraic structure of networks of intersecting planar curves in the ideal noiseless case. There are several ways to describe this structure. Our formulation is similar to Rothwell et al. [161, 163] and Bouguet et al. [14].

Consider a surface on which N planar curves $\Gamma_1, \dots, \Gamma_N$, lying on N planes $\pi_1, \dots, \pi_N$, are marked. Each curve Γ_i is just a set of points on a common plane π_i, and does not need to be continuous (our curves here are just a collection of co-planar points). For now assume an image is taken under orthographic projection, and k intersection points (x_{ij}, y_{ij}) between the projections of the curves Γ_i and Γ_j onto the image are identified. The setting is illustrated in figure 2.2. Our input is the intersection points, together with their association to the intersecting curves. The goal is to recover the planes and compute the depth along the curves. Note that two curves may not intersect in the image or they may intersect several times. For simplicity we omit the intersection multiplicity index. If two curves share a straight line segment, we pick its two endpoints as intersection points. When more than two curves intersect at a point, we consider them as pairs in a cyclic order.

Assuming the planes do not contain the projection direction (an edge-on plane provides no information and should be ignored), we can parameterize plane π_i as

$$z_i(x, y) = a_i x + b_i y + d_i \; . \tag{2.1}$$

A 2D intersection point in the image corresponds to a 3D intersection on the surface, which allows us to eliminate the unknown depths:

$$\begin{aligned} & z_i(x_{ij}, y_{ij}) - z_j(x_{ij}, y_{ij}) = \\ & (a_i - a_j)x_{ij} + (b_i - b_j)y_{ij} + (d_i - d_j) = 0 \; . \end{aligned} \tag{2.2}$$

The last equation has another geometric interpretation. Planes π_i and π_j intersect at a 3D line whose projection onto the image is the image line L_{ij} defined by

$$(a_i - a_j)x + (b_i - b_j)y + (d_i - d_j) = 0 \; . \tag{2.3}$$

Equation (2.2) simply means that (x_{ij}, y_{ij}) is on L_{ij}. Collecting equations (2.2) for all visible intersection points results in a homogeneous linear system:

$$\mathbf{A}\mathbf{v} = 0 \; , \tag{2.4}$$

where $\mathbf{v} = (a_1, \ldots, a_N, b_1, \ldots, b_N, d_1, \ldots, d_N)^T$ is a vector collecting the planes' parameters and $\mathbf{A}$ is a sparse matrix whose rows contain $x_{ij}, -x_{ij}, y_{ij}, -y_{ij}, 1, -1$ at the appropriate columns. Any vector $\mathbf{v}$ defines surface curves $\{\Gamma_i\}$ by the back-projection of the image curves onto their planes.

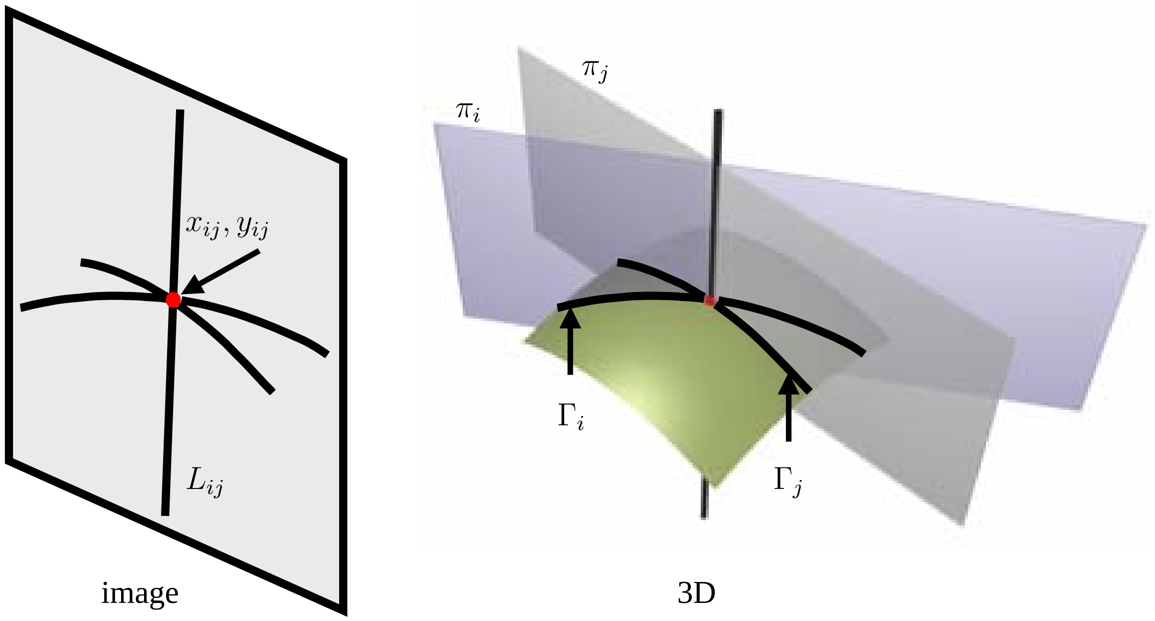


Figure 2.2: Intersecting planar curves in 3D and their projection.

Sugihara already noticed that the problem has similar structure under perspective projection. In fact, the original formulation of Falk [42] was for the perspective case. Following standard conventions, the relation between image coordinates x, y to world coordinates X, Y in perspective projection with focal length f is $x = fX/Z, y = fY/Z$. A plane π_i which does not pass through the camera center is parameterized by

$$\begin{aligned} & a_i X + b_i Y + c_i Z(X, Y) = \\ & (a_i x/f + b_i y/f + c_i) Z(X, Y) = 1 \; . \end{aligned} \tag{2.5}$$

The intersections between pairs of curves yield homogenous linear equations analogous to (2.2)

$$\begin{aligned} & 1/Z_i(x_{ij}, y_{ij}) - 1/Z_j(x_{ij}, y_{ij}) = \\ & (a_i - a_j) x_{ij}/f + (b_i - b_j) y_{ij}/f + (c_i - c_j) = 0 \; . \end{aligned} \tag{2.6}$$

In the perspective case we divide by f the entries x_{ij}, y_{ij} in the matrix $\mathbf{A}$, which we denote as $\mathbf{A}_f$. Collecting equations (2.6) for all visible intersection points results in a homogeneous linear system analogous to (2.4):

$$\mathbf{A}_f \mathbf{v}_f = 0 \; . \tag{2.7}$$

Solutions to 2.7 can be substituted in 2.5 to give the depths $Z(X,Y) = \frac{1}{a_i x/f + b_i y/f + c_i}$. The orthographic and perspective system are similar, except that in the perspective case the solution is for inverse-depths.

It is sometimes possible to reduce the number of variables in the systems (2.4) and (2.7). For instance, it might be known that some planes are parallel to the ground plane, parallel to each other (e.g. in structured light), or intersect at a single point [14]. It might also be possible to estimate the normal of a plane by finding its vanishing line in the image [144, 175, 182]. In our analysis here we consider the most general case of $3N$ variables.

Under orthographic projection, the vector $\mathbf{v}$ of true planes is a solution to (2.4) and hence in the null space of $\mathbf{A}$, which we denote $\mathrm{Null}(\mathbf{A})$. However, $\mathrm{Null}(\mathbf{A})$ also contains *trivial solutions*, which place all curves in the same arbitrary plane. Observe that (2.2) has a *basic trivial subspace* of solutions spanned by

$$\mathbf{v}_1 = \frac{(1_N, 0_{2N})^T}{\sqrt{N}}, \mathbf{v}_2 = \frac{(0_N, 1_N, 0_N)^T}{\sqrt{N}}, \mathbf{v}_3 = \frac{(0_{2N}, 1_N)^T}{\sqrt{N}}, \tag{2.8}$$

where we use the notation c_k to denote a k-vector whose entries are all c. For any non-flat solution $\mathbf{v}$ to (2.4) there is a 4D subspace of ambiguous solutions, since any linear combination of $\mathbf{v}$,$\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$ is also a solution to (2.4). This is known as Generalized Bas-Relief (GBR) ambiguity (the ordinary bas-relief ambiguity in an old depth-compression sculpting technique [207]). The intuitive interpretation of the GBR is that “adding a plane” to all points on the surface, or multiplying the depths by a constant, preserve co-planarity relationships and is invisible to the observer. A similar ambiguity occurs in the perspective case, and called Generalized Perspective Bas-Relief (GPBR). The GBR ambiguity was already known to Sugihara, although the terms GBR and GPBR became common later [7, 109]. This ambiguity is common to several uncalibrated depth cues, such as shadows, photometric stereo, and affine structure from motion from two views [102]. Interestingly, in human perception studies of pictorial relief, Koenderink et al. [106] found that variations in 3D perception of the same picture by different observers can be attributed to the orthographic GBR ambiguity.

The GBR and GPBR are special affine and projective transformations of 3D space. A GBR transformation can be written as

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ a & b & c \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} \quad , \tag{2.9}$$

for some a, b, c, d. Similarly, the GPBR can be written in homogeneous coordinates as

$$\begin{pmatrix} X' \\ Y' \\ Z' \\ W' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad . \tag{2.10}$$

Note that the GPBR scales points along their line of sight to the center of projection. The movement of points along their line of sight is unnoticed in the projected image. This is analogous to the operation of the orthographic GBR, which moves points in the direction of the z-axis.

To understand how the GPBR operates on planes, consider a 3D point $(X_0, Y_0, Z_0)^T$ on some plane $a_1X + b_1Y + c_1Z = 1$. The homogeneous coordinates of this point can be written as $(X_0, Y_0, Z_0, 1)^T = (X_0, Y_0, Z_0, a_1X_0 + b_1Y_0 + c_1Z_0)^T$. Applying the GPBR transformation (2.10) we get

$$\begin{aligned}\begin{pmatrix} X' \\ Y' \\ Z' \\ W' \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & d \end{pmatrix} \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ a_1X_0 + b_1Y_0 + c_1Z_0 \end{pmatrix} \\ &= \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ (a+a_1d)X_0 + (b+b_1d)Y_0 + (c+c_1d)Z_0 \end{pmatrix} \quad . \end{aligned} \tag{2.11}$$

If $W' \neq 0$, the point $(X', Y', Z', W')^T$ is on the plane

$$(a+a_1d)X + (b+b_1d)Y + (c+c_1d)Z = 1 \quad . \tag{2.12}$$

Thus, under a GPBR transformation, a vector $\mathbf{v} = (a_1, \ldots, a_N, b_1, \ldots, b_N, d_1, \ldots, d_N)^T$ of planes' parameters is transformed to $a\sqrt{N}\mathbf{v}_1 + b\sqrt{N}\mathbf{v}_2 + c\sqrt{N}\mathbf{v}_3 + d\mathbf{v}$.

It is worth pointing out that there are several alternative formulations of co-planarity constraints. These formulations are mathematically equivalent to ours in the noiseless case, and should produce similar results on small line drawings of polyhedra. However, they are not equivalent computationally in structured light scenarios in terms of size and sparsity of the matrices involved.

- The original formulation by Sugihara [183] had all the variables $(a_1, \ldots, a_N, b_1, \ldots, b_N, d_1, \ldots, d_N, z_1, \ldots, z_k)$ in the linear system. In a structured light scenario we might have $k = O(N^2)$ intersection points, while only $3N$ parameters are required to represent the planes. This could be a problem for large N if we use dense solvers.

- Lipson and Shpitalni [119] formed co-planarity constraints by plane-fitting. We can use this idea (in a different way than theirs) to derive linear constraints on the depths $\mathbf{z} = (z_1, .., z_m)^T$ of m co-planar points as follows. For co-planar points there exist a, b, d such that
$$\begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 1 \end{bmatrix} \begin{pmatrix} a \\ b \\ d \end{pmatrix} = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} = \mathbf{z} \,. \tag{2.13}$$
Let
$$\mathbf{P} = \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & 1 \end{bmatrix} . \tag{2.14}$$
Then
$$\mathbf{P}^+\mathbf{z} = \begin{pmatrix} a \\ b \\ d \end{pmatrix} , \tag{2.15}$$
$$(\mathbf{I} - \mathbf{P}\mathbf{P}^+)\mathbf{z} = 0, \tag{2.16}$$
where $\mathbf{P}^+$ is the pseudo-inverse of $\mathbf{P}$. If we construct $\mathbf{S}_i = (\mathbf{I} - \mathbf{P}_i\mathbf{P}_i^+)$, where $\mathbf{P}_i$ contains the points of each planar face, then we can pad these matrices with zeros such that the matrix $\mathbf{S} = \begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_N \end{bmatrix}$ represents all the co-planarity constraints, i.e. $\mathbf{S}\mathbf{z} = 0$, where $\mathbf{z} = (z_1, .., z_k)^T$. In this form, not only the size of the matrix $\mathbf{S}$ can be $O(k \times k)$, but also $\mathbf{S}$ can be denser than our matrix $\mathbf{A}$.

- Sparr [180] and Heyden [72] proposed sparse constraints on the depths of co-planar points using barycentric representations. For each planar face we can choose three non-collinear points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ as a basis. Any other point on that face with image coordinates (x_4, y_4) can be represented as

$$\begin{pmatrix} x_4 \\ y_4 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \alpha\left(\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} - \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) + \beta\left(\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} - \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) \tag{2.17}$$

This can be rearranged as

$$\begin{aligned} & c_1\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + c_2\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + c_3\begin{pmatrix} x_3 \\ y_3 \end{pmatrix} + c_4\begin{pmatrix} x_4 \\ y_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ & c_1 + c_2 + c_3 + c_4 = 0. \end{aligned} \tag{2.18}$$

The barycentric representation is invariant to affine transformations and the GBR in particular. When the face is lifted from the image plane to 3D by some GBR transformation, the linear constraint $c_1z_1 + c_2z_2 + c_3z_3 + c_4z_4 = 0$ still holds. The constraints for all points over all planes can be collected into a matrix $\mathbf{S}$, so that $\mathbf{Sz} = 0$ as before. Crapo [34] obtained a similar matrix $\mathbf{S}$, where instead of solving (2.17) numerically for α and β, his matrix entries are expressed using determinants. Although $\mathbf{S}$ has the same order of nonzero elements as $\mathbf{A}$, it is possibly of size $O(k \times k)$.

- Interestingly, the question whether a line drawing can be realized as a non-flat 3D polyhedron is connected to mechanical rigidity of networks of planar faces connected by hinges. A linear system derived from kinematic considerations was proposed by Ros and Thomas [158, 159]. In their system, the variables are the infinitesimal angular velocities of the hinges that would lift the network from the image plane into a non-flat 3D configuration. While this scheme is similar to the previous ones for polyhedra, it relies on straight segments in the line drawing. However, in structured light we might have a single intersection point between two curves in the image, so no hinge can be formed.

Chapter 3

A Linear Escape from Flatland

3.1 Introduction

In this chapter we revisit the problem of recovering 3D shape from the projection of 3D planar curves. This problem is strongly motivated by perception studies, and has applications in single-view modeling and uncalibrated structured light. As we saw in the previous chapter, systems of intersecting planar curves lead to linear systems of equations. Here we take a closer look at the non-trivial solutions of such systems.

Previous work on linear systems from planar curves was divided between work on planar faces and planar cross sections. Planar faces of polyhedral objects were studied by Sugihara [183] and many others. Linear systems arising from planar cross sections have been examined in structured light [14] and Straight Homogeneous Generalized Cylinders (SHGC) [196]. Note that the planar cross sections of an SHGC need not necessarily be visible in the image. In some cases they could be estimated from the silhouette [192]. We stress that the same analysis applies to scenes having both types of planar curves. This point has also been emphasized in independent work [53, 54, 89–91] that followed our paper [41].

The algorithm we propose looks for a solution that minimizes the algebraic error of the linear system, under the constraint that a geometrically meaningful measure of non-planarity is held constant. This formulation has two advantages. First, it is less sensitive to noise than a direct least-squares solution. Second, our cost function is invariant to changes in focal length. Thus, given a system of intersecting curves with little amount of noise, our method can determine whether it is possible that all curves are actually planar in a non-trivial 3D configuration. We also show examples where the same method produces plausible surfaces from the projection of parallel (non-intersecting) planar cross sections.

3.2 Non-trivial solutions in noiseless systems

Before getting into solving the system $\mathbf{Av} = 0$ (2.4) we consider whether multiple linearly-independent solutions are possible. Ideally our system would have a 4D solution space: three dimensions are due to the GBR ambiguity and the fourth comes from the true shape. One may wonder whether systems with more independent non-trivial solutions exist. The dimension clearly depends on the observed pattern of intersections. For instance, disjoint components of curves leave some degrees of freedom. If the image is of a triangular mesh, we can place any of its vertices freely in depth. Rothwell and Stern [161] demonstrated polyhedra with more than four degrees of freedom but relatively few intersections. However, for highly-connected networks of curves the system will typically be over-determined. For example, in a structured light scenario we may have $O(N^2)$ equations in $3N$ unknowns. Although extremely rare, there exist highly-connected systems with $O(N^2)$ intersections that have a subspace of non-trivial solutions whose dimension is greater than one, as shown in figure 3.1.

Next we show how to artificially construct systems with $O(N^2)$ intersection points that have a 2-dimensional non-trivial subspace. This question is not completely trivial. One cannot manipulate the entries of the matrix $\mathbf{A}$ arbitrarily to lower the rank of the matrix, because the matrix has a special structure: on each row we have $x_{ij}, -x_{ij}, y_{ij}, -y_{ij}, 1, -1$. However, it can be easily achieved with a geometric construction. Let $\{\pi_i^1\}$,$\{\pi_i^2\}$ be any two sets of N planes in general position. For each set of planes, we can intersect pairs of planes in 3D and find the

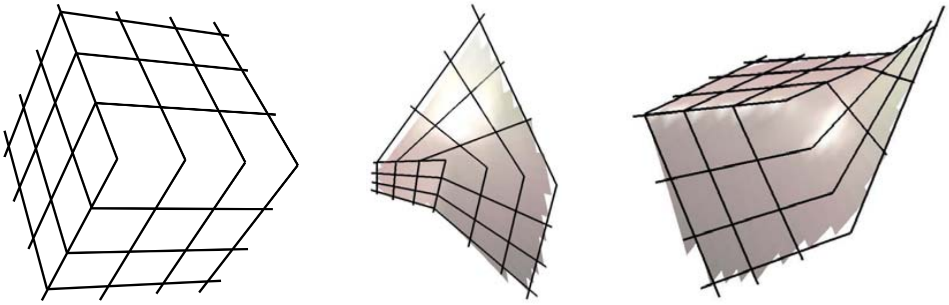


Figure 3.1: Multiple interpretations. In the left drawing, each "L" shaped curve is assumed to be planar. The drawing can be interpreted as a cube or as having doubly-ruled faces. In this particular example, there is a 7-dimensional solution subspace, since each vertex of the cube can be placed arbitrarily in depth.

projection of the intersection lines in the image, as defined by (2.3). Denote by $\{L^1_{ij}\}$,$\{L^2_{ij}\}$ these image lines for each set of planes. Set (x_{ij}, y_{ij}) to be the intersection point of L^1_{ij} and L^2_{ij}. By construction, the two sets of planes $\{\pi^1_i\}$,$\{\pi^2_i\}$ are solutions to the system defined by this choice of (x_{ij}, y_{ij}). Having a 3-dimensional non-trivial subspace is much more accidental, since for each intersection the lines L^1_{ij},L^2_{ij},L^3_{ij} must intersect at a point (or two lines coincide).

3.3 Solving noisy systems

We now turn to solving the system in practice. Sugihara [183] already noticed what he called the superstrictness problem: the linear system is typically over-determined and has no exact non-trivial solution in presence of noise. There are two general ways to deal with this problem.

The first approach assumes that the curves are exactly planar in the world and therefore the measured positions in the image of the intersection points need to be corrected. Several methods fall into this category. Sugihara and Imai [183] look for a subset of the equations such that if we drop these equations, the system has a solution even if the positions of the intersection points are slightly perturbed (generically reconstructible system). This approach does not spread the error evenly [160]. Shimshoni and Ponce [173] derived a linear programming relaxation. Their method provides a necessary but not sufficient condition for non-trivial 3D realization of the curves. Ros and Thomas [160] proposed a nonlinear optimization algorithm based on a special parameterization of all exact solutions. The parameterization is derived from a resolvable sequence, which is an ordering of the vertices and faces such that there is always a way to place the vertices and faces one by one in 3D without contradicting previous placements. However, as they discuss, there are systems for which a resolvable sequence does not exist. Heyden [72] examined modifying the matrix of the system using SVD on any inconsistent subset of equations. While his method is not guaranteed to work for all systems, it was probably the first purely numerical method for the superstrictness problem.

The second general approach to the superstrictness problem assumes the positions of the intersection points are correct ($\mathbf{A}$ is fixed), and accepts the compromise that the curves will not be exactly planar in 3D. Since curves that are exactly planar are uncommon in nature, we may consider planarity as a simplifying assumption that could be violated. A clear advantage of this approach is that numerical linear algebra methods can be applied. In contrast, the approach of correcting the line drawing is more combinatorial in nature and generally is a hard problem. Previous work along this line include Grimstead and Martin [62, 198], Sturm and Maybank [182], Bouguet et al. [14], and Ros and Thomas [159]. Our work follows this direction.

3.3.1 The trivial subspace

We begin our investigation with a closer look at the trivial subspace. It might be that the vectors $\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$ from equation (2.8) do not span the full algebraic subspace of flat solutions. This happens when there is a curve whose intersection points are along a straight line (in particular a curve with less than three intersections), as illustrated in figure 3.2. Assuming the planes are not edge-on implies that a straight curve in the image is straight in 3D. Note that Zero Gaussian Curvature (ZGC) surfaces [195] are a special case of surfaces that contain lines.

To characterize the complete space of flat solutions we need a measure of surface flatness. A natural approach is to pick a set of points on the surface, fit a plane with linear regression, and

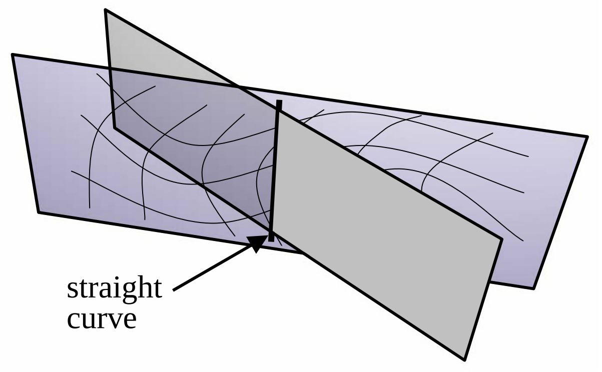


Figure 3.2: A trivial solution can embed all curves into the same geometric plane while assigning the straight line a different plane. Since all the intersection points are on the same geometric plane, there are no gaps between the curves at the intersection points.

measure the residual error. The error will be zero whenever all points are coplanar. Although we don't know the depths at the points, we can express these depths as a linear function of $\mathbf{v}$. Let (x_i, y_i) be a set of k representative points on the curves (e.g. the intersection points). Let $\mathbf{Z}$ be a matrix that reverse-projects the points onto their planes. The rows of $\mathbf{Z}$ contain $x_i, y_i, 1$ at the appropriate columns, so that $\mathbf{Zv} = (z_1, \ldots, z_k)^T$, where $(z_1, \ldots, z_k)^T$ are the depths of the planes at the points (x_i, y_i) according to equation (2.1). The geometry is illustrated in figure 3.3, where for simplicity it shows the intersections of lines in 2D instead of planes in 3D. Define the matrix $\mathbf{C}$ by

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_k & y_k & 1 \end{bmatrix} \quad , \\ \mathbf{C} &= \left(\mathbf{I} - \mathbf{P}\mathbf{P}^{+}\right)\mathbf{Z}/\sqrt{k} = \left(\mathbf{Z} - \mathbf{P}\left(\mathbf{P}^{+}\mathbf{Z}\right)\right)/\sqrt{k} \quad , \end{aligned} \tag{3.1}$$

where $\mathbf{P}^{+}$ is the pseudo-inverse of $\mathbf{P}$. Then $\|\mathbf{Cv}\|^2$ is the averaged squared norm of the deviation in the z direction of the points (x_i, y_i, z_i) from the best fitting plane obtained by plane-fitting linear regression. The residual of this regression is the solution to the least-squares

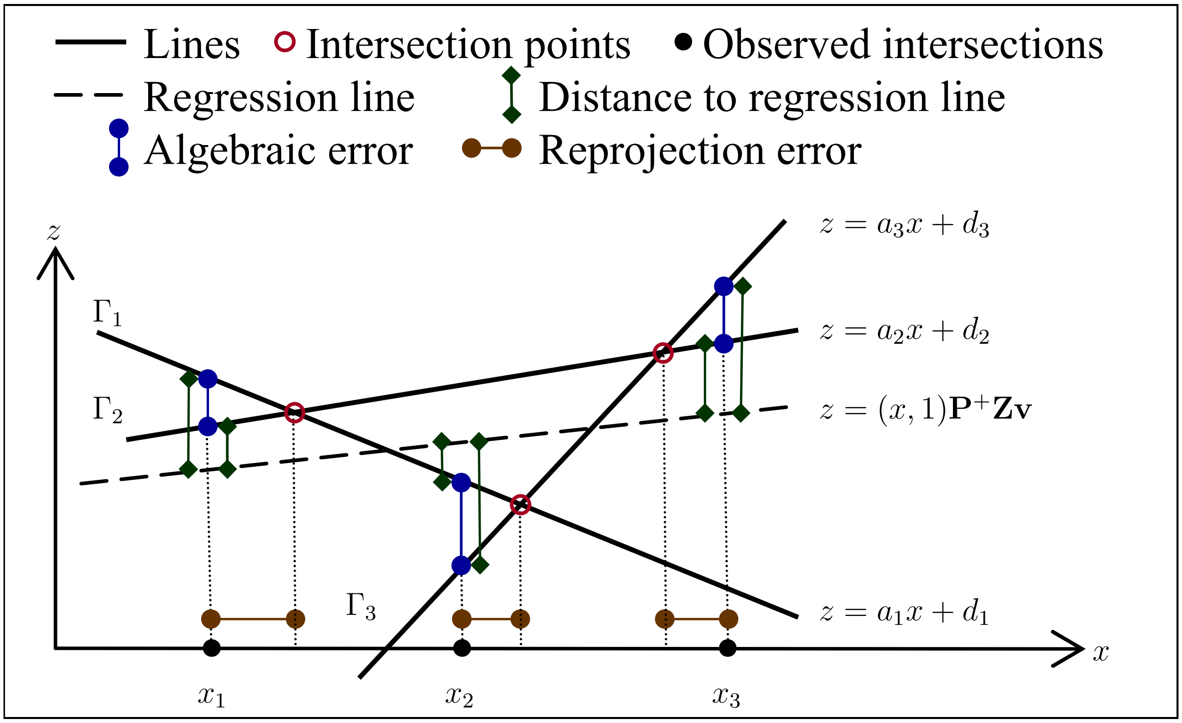


Figure 3.3: System of three intersecting lines $(\Gamma_1, \Gamma_2, \Gamma_3)$ in 2D. The image is formed by projection on the x-axis. Here $\mathbf{Z} = \begin{bmatrix} x_1 & 0 & 0 & 1 & 0 & 0 \\ 0 & x_1 & 0 & 0 & 1 & 0 \\ x_2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & x_2 & 0 & 0 & 1 \\ 0 & 0 & x_3 & 0 & 0 & 1 \\ 0 & x_3 & 0 & 0 & 1 & 0 \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$, $\mathbf{P} = \begin{bmatrix} x_1 & 1 \\ x_1 & 1 \\ x_2 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_3 & 1 \end{bmatrix}$.

problem

$$\min \frac{1}{k} \left\| \mathbf{Zv} - \mathbf{P} \begin{pmatrix} a \\ b \\ d \end{pmatrix} \right\|^2 , \tag{3.2}$$

which is achieved by $(a, b, d)^T = \mathbf{P}^{+}\mathbf{Zv}$. Therefore, the residual in equation (3.2) equals $\frac{1}{k}\|(\mathbf{Z} - \mathbf{PP}^{+}\mathbf{Z})\mathbf{v}\|^2 = \|\mathbf{Cv}\|^2$.

We define the *trivial subspace* to be the null space of $\mathbf{C}$. All vectors in the trivial subspace correspond to coplanar points (x_i, y_i, z_i), including cases with straight curves as in figure 3.2. There are three degrees of freedom in choosing this plane, but the dimension of $\mathrm{Null}(\mathbf{C})$ could

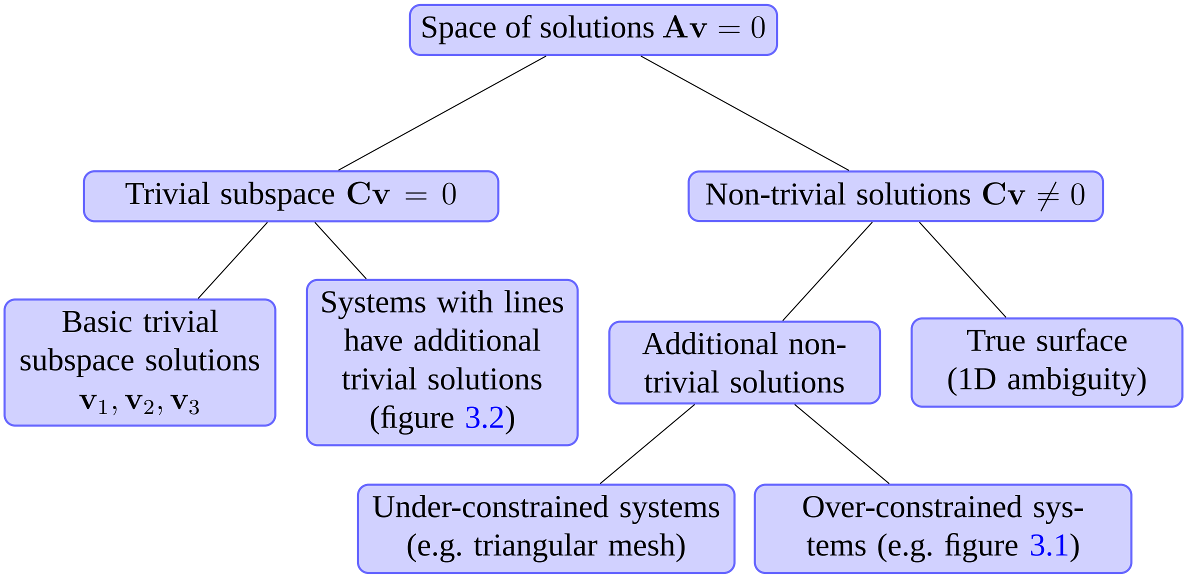

Space of solutions $\mathbf{A}\mathbf{v} = 0$
Trivial subspace $\mathbf{C}\mathbf{v} = 0$
Non-trivial solutions $\mathbf{C}\mathbf{v} \neq 0$
Basic trivial subspace solutions $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$
Systems with lines have additional trivial solutions (figure 3.2)
Additional non-trivial solutions
True surface (1D ambiguity)
Under-constrained systems (e.g. triangular mesh)
Over-constrained systems (e.g. figure 3.1)

Figure 3.4: Categories of solutions to noiseless linear systems of intersecting planar curves. A solution to the system $\mathbf{A}\mathbf{v} = 0$ can be a linear combination of trivial and non-trivial solutions. A system should have at least a four-dimensional solution space (GBR ambiguity). Additional linearly-independent solutions are also possible.

be larger. A categorization of solutions in the noiseless case is shown in figure 3.4.

The trivial subspace has degrees of freedom which could be set with additional information (e.g. depth of some known points). In absence of additional information, under orthography one may get a convincing qualitative shape by picking a solution which is orthogonal to the trivial subspace (under perspective this might place observed points behind the camera). This choice is natural since while the planar solutions are common to any set of curves, the non-trivial component is specific to the observed curves, i.e. to the specific problem data. Related ideas appear in [14,161]. Interestingly, solutions orthogonal to 1_N occur in diverse applications such as graph Laplacians or graph drawing [65,107]. Next we provide some statistical support for this choice.

Assume a set of random planes is viewed orthographically from a random direction, or simply that the planes have random uniform orientations. What is then the distribution of the basic trivial component? Recall that $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in equation (2.8) are an orthonormal basis

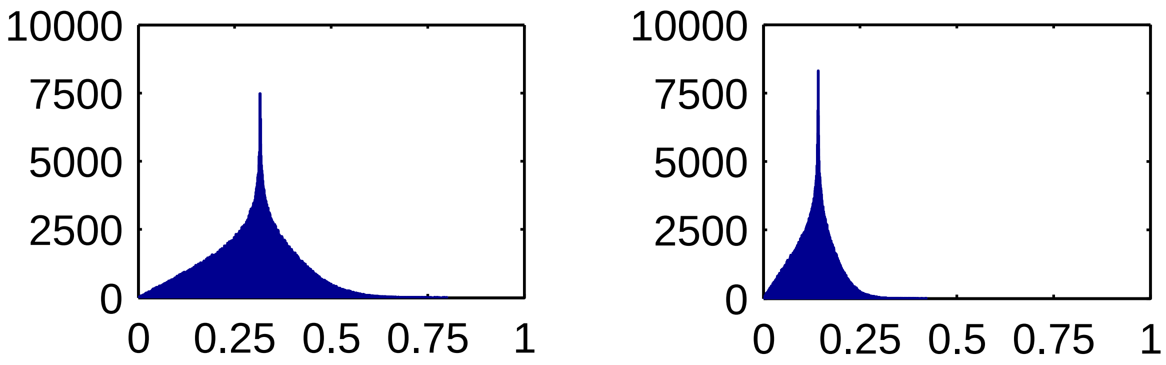


Figure 3.5: Histograms of (3.4) for random planes (10^6 trials). Left: $N = 10$ planes, $mean = 0.297$, $std = 0.108$. Right: $N = 50$, $mean = 0.132$, $std = 0.05$.

vectors for the basic trivial subspace. The relative magnitude of the basic trivial component is

$$\frac{\left\| \left(\mathbf{v}^T\mathbf{v}_1, \mathbf{v}^T\mathbf{v}_2, \mathbf{v}^T\mathbf{v}_3\right) \right\|}{\|\mathbf{v}\|} . \tag{3.3}$$

Since absolute depth is lost in orthographic view, we may set $\mathbf{v}^T\mathbf{v}_3 = \frac{1}{\sqrt{N}}\sum d_i = 0$. An upper bound on (3.3) is

$$\frac{\|(\sum a_i, \sum b_i)\|}{\sqrt{N}\|(a_1, \ldots, a_N, b_1, \ldots, b_N)\|} . \tag{3.4}$$

When the orientations of the planes are uniformly distributed, the directions of the unscaled normals are $(n_i^x, n_i^y, -n_i^z)$, where $n_i^x, n_i^y, n_i^z \sim \mathcal{N}(0,1)$. In the form of (2.1), $a_i \sim n_i^x/n_i^z$, $b_i \sim n_i^y/n_i^z$. The coefficients of the planes are Cauchy distributed. The sum of Cauchy variables is also distributed Cauchy, has a mode at zero but its mean is undefined. Therefore, we cannot apply the central limit theorem (CLT) to the absolute magnitude of the trivial component, $\frac{1}{\sqrt{N}}\sum a_i$ or $\frac{1}{\sqrt{N}}\sum b_i$, and expect that as $N \to \infty$ their distributions will converge to the normal distribution. However, the relative magnitude of the basic trivial component in (3.3) is always bounded by 1. The empiric histogram of (3.4) is shown in figure 3.5. From this empiric distribution we conclude that under the assumption of large number of uniformly distributed plane normals, it is probable that the magnitudes of $\mathbf{v}^T\mathbf{v}_1$ and $\mathbf{v}^T\mathbf{v}_2$ are small relative to the magnitude of $\mathbf{v}$.

3.3.2 Proposed method for orthographic systems

In the presence of noise the trivial (planar) solutions are still exact, but in general there are no additional exact solutions. Geometrically this means there may be depth gaps between the planes at (x_{ij}, y_{ij}). This is shown in figure 3.3 as the algebraic error. A standard approach is to minimize the norm of the residuals of (2.4), i.e. looking for a vector $\mathbf{v}$ such that $\|\mathbf{Av}\|$ is small. However, when the points are nearly planar $\|\mathbf{Av}\|$ is small for any set of curves. Thus, we want at the same time to keep the points (x_i, y_i, z_i) away from a common plane. The condition $\|\mathbf{Cv}\| = 1$ holds the points away from their best fitting plane.

Recall that $\mathrm{Null}(\mathbf{C})$ is made of the trivial solutions that can be fit with a single plane. From (3.1,3.2), $\mathbf{Cv} = 0$ implies $\mathbf{Zv} = \mathbf{P}(a, b, d)^T$ for some plane parameters a, b, d. Hence, $z_i(x_{ij}, y_{ij}) = z_j(x_{ij}, y_{ij}) = ax_{ij} + by_{ij} + d$. By definition, that means $\mathbf{Av} = 0$. Therefore, $\mathrm{Null}(\mathbf{C}) \subseteq \mathrm{Null}(\mathbf{A})$, even when $\mathbf{A}$ is noisy. We may assume for now that $\mathbf{v}$ is orthogonal to $\mathrm{Null}(\mathbf{C})$, since any component of $\mathbf{v}$ in the trivial subspace does not affect $\|\mathbf{Av}\|$. The trivial component can be added later if additional information is provided. We pose (2.4) as

$$\operatorname*{argmin}_{\mathbf{v}} \|\mathbf{Av}\| \ \text{ s.t. } \|\mathbf{Cv}\| = 1 \ , \ \mathbf{v} \perp \mathrm{Null}(\mathbf{C}) \ . \tag{3.5}$$

To solve (3.5), let $\mathbf{C} = \mathbf{U}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^T$ be the SVD of $\mathbf{C}$. Define $\mathbf{V}$ and $\mathbf{D}$ by removing the columns of $\tilde{\mathbf{V}}$ and rows of $\tilde{\mathbf{D}}$ that correspond to singular values smaller than ε. The removed columns of $\tilde{\mathbf{V}}$ span the trivial subspace, which is at least three-dimensional. The columns of $\mathbf{V}$ form a basis for the orthogonal complement of the trivial subspace. Writing $\mathbf{v} = \mathbf{VD}^{-1}\mathbf{w}$, and using the fact that $\mathbf{U}$ is orthogonal, the problem becomes

$$\operatorname*{argmin}_{\mathbf{w}} \|\mathbf{AVD}^{-1}\mathbf{w}\| \ \text{ s.t. } \|\mathbf{UDV}^T\mathbf{v}\| = \|\mathbf{w}\| = 1 \ . \tag{3.6}$$

The procedure is summarized in algorithm 1. In this algorithm, $\mathbf{w}$ essentially picks a solution in the $\mathbf{V}$ basis. The matrix $\mathbf{D}^{-1}$ weights the columns of $\mathbf{V}$ so that vectors closer to a flat solution get higher cost. The singular value associated with $\mathbf{w}$ is the minimum value of (3.5). For a system with a little amount of noise, we expect its value to be close to zero, unless the true surface is a trivial solution.

Algorithm 1 Solving a system of intersecting planar curves.

```
1: Form the matrices $\mathbf{A}, \mathbf{Z}, \mathbf{P}$ from the intersection points.
2: Set $\mathbf{C} = \left(\mathbf{Z} - \mathbf{P}\left(\mathbf{P}^{+}\mathbf{Z}\right)\right)/\sqrt{k}$.
3: Compute the SVD $\mathbf{C} = \mathbf{U}\tilde{\mathbf{D}}\tilde{\mathbf{V}}^{T}$.
4: Form $\mathbf{V}$ and $\mathbf{D}$ by removing the columns of $\tilde{\mathbf{V}}$ and rows of $\tilde{\mathbf{D}}$ that correspond to singular
   values smaller than $\varepsilon$.
5: Compute the SVD of $\mathbf{A}\mathbf{V}\mathbf{D}^{-1}$, and let $\mathbf{w}$ be the last right singular vector.
6: Return $\mathbf{v} = \mathbf{V}\mathbf{D}^{-1}\mathbf{w}$.
```

3.3.3 Perspective systems

In the previous chapter we met the orthographic system $\mathbf{A}\mathbf{v} = 0$ (2.4) and the perspective system $\mathbf{A}_f\mathbf{v}_f = 0$ (2.7). Sugihara [183] observed that the orthographic system has an exact non-trivial solution if and only if the perspective system with the same intersection points has an exact solution. His result implies that given a noiseless systems of curves, one can test whether it is possible that the curves are indeed planar in a non-trivial configuration, without knowing the focal length of the camera. It is always enough to check whether a solution exists assuming orthographic projection. We extend this result to noisy systems by showing that the error level measured in our formulation for noisy systems is independent of the focal length. Note that in general, the singular values of the matrix $\mathbf{A}_f$ from (2.7) are not invariant to changes of f.

In the perspective case we divide by f the entries x_{ij}, y_{ij}, x_i, y_i in the matrices $\mathbf{A}, \mathbf{Z}, \mathbf{P}$, which we denote as $\mathbf{A}_f, \mathbf{Z}_f, \mathbf{P}_f$, and compute $\mathbf{C}_f$ similar to (3.1). As in the orthographic case, the matrix $\mathbf{C}_f$ performs a plane-fit. Instead of fitting a plane to the 3D points (X_i, Y_i, Z_i), it is algebraically more convenient to define $\mathbf{C}_f$ such that it measures deviation from planarity for the transformed points $(\frac{X_i}{Z_i} = \frac{x_i}{f}, \frac{Y_i}{Z_i} = \frac{y_i}{f}, \frac{1}{Z_i})$. Note that if the transformed points are on a plane $a\frac{X_i}{Z_i} + b\frac{Y_i}{Z_i} + c = \frac{1}{Z_i}$, then the 3D points are on the plane $aX_i + bY_i + cZ_i = 1$. With these definitions we can solve

$$\underset{\mathbf{v}_f}{\operatorname{argmin}} \|\mathbf{A}_f \mathbf{v}_f\| \ \text{ s.t. } \|\mathbf{C}_f \mathbf{v}_f\| = 1 \ . \tag{3.7}$$

We show below that for any focal length f, the smallest singular values obtained from solving equation (3.7) are the same, and equal those of the orthographic system. With any solution vector $\mathbf{v} = (a_1, \ldots, a_N, b_1, \ldots, b_N, d_1, \ldots, d_N)^T$ to the orthographic system we can associate a solution $\mathbf{v}_f$ to the perspective system

$$\mathbf{v}_f = (fa_1, \ldots, fa_N, fb_1, \ldots, fb_N, d_1, \ldots, d_N)^T \ . \tag{3.8}$$

By direct computation we get $\mathbf{A}_f \mathbf{v}_f = \mathbf{A}\mathbf{v}$, $\mathbf{Z}_f \mathbf{v}_f = \mathbf{Z}\mathbf{v}$, $\mathbf{P}_f \mathbf{P}_f^+ = \mathbf{P}\mathbf{P}^+$, and $\mathbf{C}_f \mathbf{v}_f = \mathbf{C}\mathbf{v}$. Thus, the minimal singular values of the systems (3.5) and (3.7) are the same. Note that if $\mathbf{v}$ is perpendicular to the basic trivial subspace, then so is $\mathbf{v}_f$. However, if $\mathbf{v}$ is perpendicular to the numerical null space of $\mathbf{C}$, there is no guarantee that $\mathbf{v}_f$ will be orthogonal to the numerical null space of $\mathbf{C}_f$. Nevertheless, since $\text{Null}(\mathbf{C}_f) \subseteq \text{Null}(\mathbf{A}_f)$, it is always possible to project out the trivial components in $\mathbf{v}_f$ without affecting the cost value. However, our argument for picking a solution orthogonal to the trivial subspace is less justified for perspective projection, since it may place some visible points behind the camera (violate the cheirality constraints). In some situations the solution must contain a trivial component.

3.3.4 Results

Demonstrations of the method to uncalibrated structured light are shown in figures 2.1 and 3.6. The curves were extracted from a video sequence of projected laser strips controlled by hand. Curves whose intersection points were nearly linear were pruned, because some of these curves are not nearly linear in 3D. The surfaces shown were computed using the orthographic model. Surface interpolation between the curves was done by Matlab, which interpolates over a trian-

Table 3.1: The five smallest singular values of $\mathbf{AVD}^{-1}$.

Fig. 2.1	Fig. 3.6	Fig. 3.7	Fig. 3.8	Fig. 3.10 without noise (A,J)	Fig. 3.10 with noise (C,K)
0.8630	1.3247	1.2936	0.3672	2.9201e-011	35.8137
2.0788	3.8083	3.3439	1.2055	43.3110	43.8159
2.5905	5.7032	4.8551	1.6912	51.2596	52.5848
4.0279	7.0677	7.6960	4.1249	52.1086	53.7654
6.2370	9.9011	9.9925	4.7813	56.0042	58.0213

gulation.

We also applied the method for single-view modeling [182]. The user creates a 3D model from a photograph by drawing line segments and linking them to planar faces. Results are shown in figures 3.7 and 3.8. Though many faces had to be defined to fix all degrees of freedom in these examples, we used nothing but grouping of segments to planes to infer the 3D shape.

Table 3.1 lists the smallest singular values of the matrix $\mathbf{AVD}^{-1}$ for various figures. For ideal noiseless systems, the smallest singular value should be zero. Generally, a large gap separating the smallest singular value from the rest indicates that the last singular vector is likely to be close to the true set of planes (up to a GBR ambiguity).

3.3.5 Comparison to previous work

Structured light is a popular scanning method, especially in controlled environments. One inconvenience of fully-calibrated structured light is that any projected strip has to be calibrated with respect to the camera. This makes it difficult to move the camera or light freely to achieve optimal scanning of the object. Hence, it is desirable to minimize the amount of calibration. The method we presented is fully uncalibrated. It proposes a shape assuming orthogonality to the trivial subspace. As we argued and demonstrated, this assumption produces a reasonable qualitative shape for randomly oriented planes. Otherwise, some known points, or orthogonality constraints, have to be used to achieve a metric reconstruction. The minimal additional

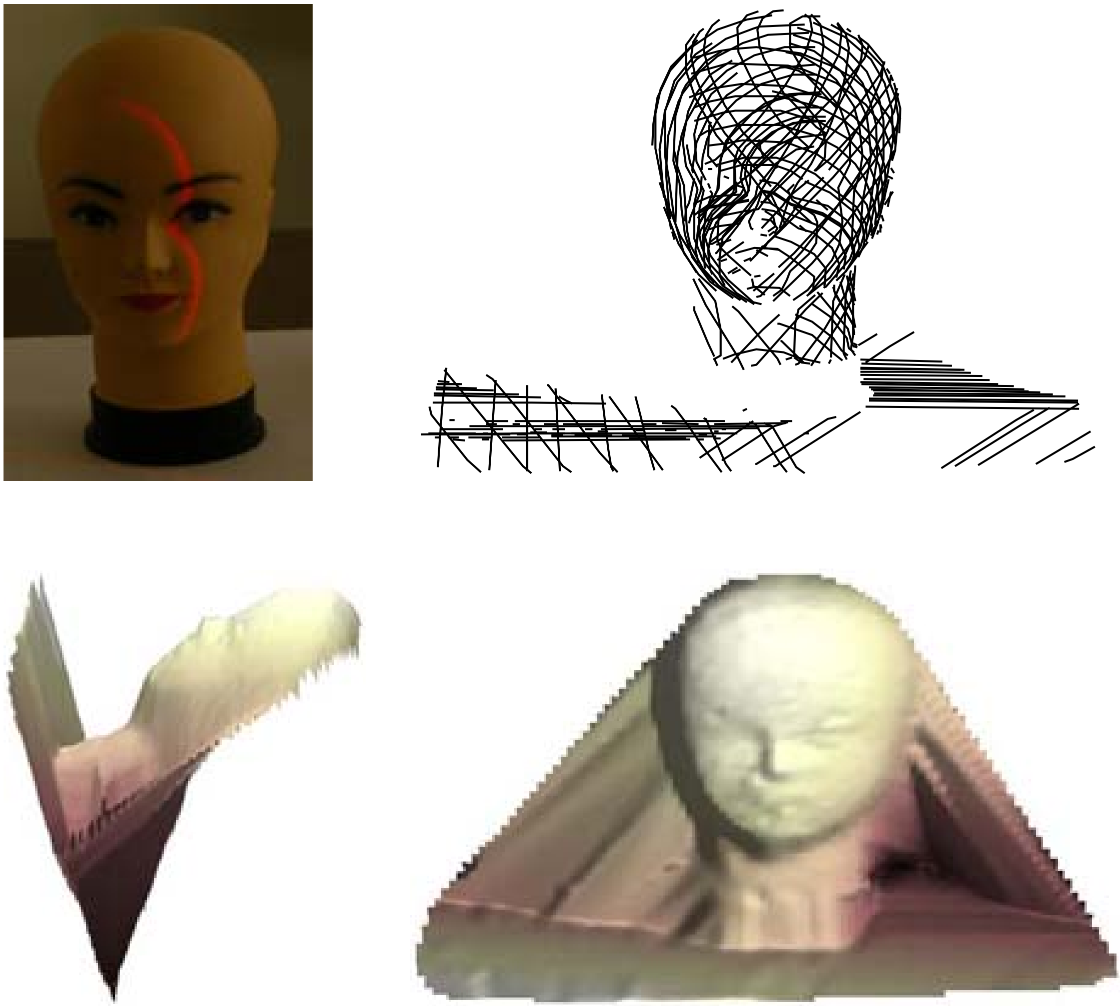


Figure 3.6: Surface reconstruction from 78 curves and 1053 intersection points. The axis of the head is not perpendicular to the table due to a non-zero trivial component in the collection of true planes.

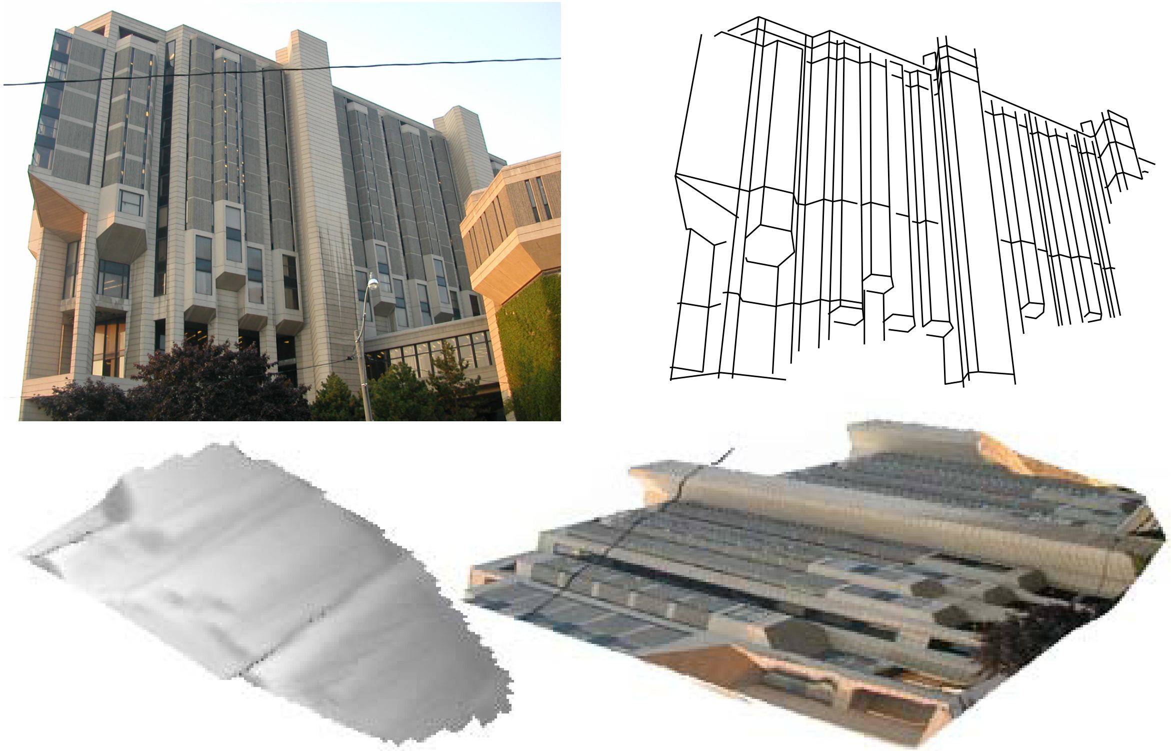


Figure 3.7: Top row: input image and manually drawn curves (54 planar faces). Bottom row: shaded and texture-mapped computed surface from different viewpoints.

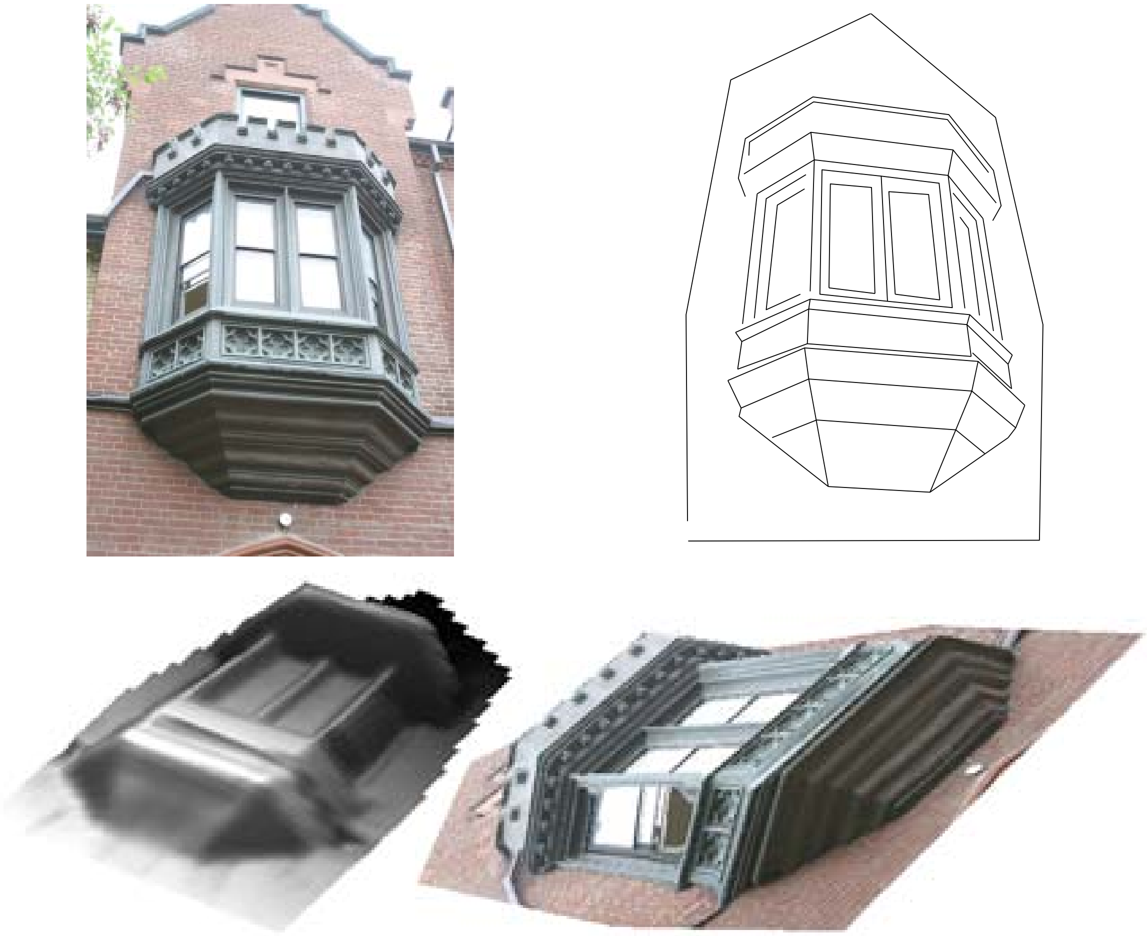


Figure 3.8: Image, curves, shaded and texture-mapped surface.

information needed is much less than what is required for calibrating every light strip. Solving a system of intersecting planes with additional calibration information yields a joint-calibration of all the planes together.

Several previous works have been attempting to do uncalibrated structured light. Chen, Gao and Chen [26] assume the projected planes form an orthogonal grid. Caspi and Werman [21] assume the planes belong to two pencils. In the setting of Bouguet et al. [14] it was assumed that all planes intersect at a known point (the light source) so each plane had only two degrees of freedom. Our formulation is more general than the previously used models for uncalibrated structured light. It does not restrict the planes in any way and thereby allows scanning any visible part of the surface.

In this subsection we derive the algorithm of Bouguet et al. [14] in a different way (their formulation makes the solution exactly orthogonal to the basic trivial subspace only when an exact solution exists), and compare it to ours. To avoid the zero solution ($\mathbf{v}_0 = 0_{3N}$) to (2.4), write

$$\underset{\mathbf{v}}{\operatorname{argmin}} \; \|\mathbf{A}\mathbf{v}\| \text{ s.t. } \|\mathbf{v}\| = 1, \; \mathbf{v} \perp \operatorname{Span}\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\} \; . \tag{3.9}$$

Essentially, one is looking for the fourth smallest right singular vector of $\mathbf{A}$. To enforce the solution to be orthogonal to the basic trivial subspace (both in the noisy and noiseless cases), we write $\mathbf{v} = \mathbf{B}\mathbf{u}$, where $\mathbf{B}$ is a matrix whose columns form an orthonormal basis to the orthogonal complement of $\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$ defined by (2.8). A possible closed-form choice is

$$\mathbf{B} = \begin{bmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} \end{bmatrix}_{3N\times(3N-3)} , \; \alpha = -\frac{1}{\sqrt{N}(1+\sqrt{N})} ,$$
$$\mathbf{E} = \begin{bmatrix} -\frac{1}{\sqrt{N}} & -\frac{1}{\sqrt{N}} & \cdots & -\frac{1}{\sqrt{N}} \\ 1+\alpha & \alpha & \cdots & \alpha \\ \alpha & 1+\alpha & \cdots & \alpha \\ \vdots & \vdots & \ddots & \vdots \\ \alpha & \alpha & \cdots & 1+\alpha \end{bmatrix}_{N\times(N-1)} \tag{3.10}$$

(a less symmetric choice for $\mathbf{E}$ is the Helmert matrix). $\mathbf{B}$ has orthonormal columns since for each column $\mathbf{E}_i$ we have $\|\mathbf{E}_i\|^2 = \frac{1}{N} + (N-2)\alpha^2 + (1+\alpha)^2 = 1$, and for $i \neq j$ $\mathbf{E}_i \cdot \mathbf{E}_j = \frac{1}{N} + 2\alpha(1+\alpha) + (N-3)\alpha^2 = 0$. In addition, $\mathbf{E}_i \cdot 1_N = -\frac{1}{\sqrt{N}} + 1 + (N-1)\alpha = 0$. Using this basis, the problem becomes

$$\underset{\mathbf{u}}{\text{argmin}} \; \|\mathbf{ABu}\| \;\text{ s.t. } \|\mathbf{Bu}\| = \|\mathbf{u}\| = 1 \; . \tag{3.11}$$

Let $\mathbf{AB} = \mathbf{UDV}^T$ be the SVD of $\mathbf{AB}$. $\mathbf{u}$ is the last column of $\mathbf{V}$ and $\mathbf{v} = \mathbf{Bu}$ is the desired solution. We can verify that except for the columns associated with the intersection points of the first curve, the other columns of $\mathbf{AB}$ are sparse and contain the same elements as $\mathbf{A}$. Entries of the form $\alpha x_{ij} - \alpha x_{ij}$ vanish, and entries of the form $(1+\alpha)x_{ij} - \alpha x_{ij}$ leave the original elements of $\mathbf{A}$. Since only the last singular vector is used, there might be iterative methods that can handle very large sparse systems faster than the complete computation of the SVD.

Comparing (3.6) to (3.11), the matrix $\mathbf{VD}^{-1}$ is replaced with $\mathbf{B}$. The methods differ in two aspects. First, excluding only the basic trivial subspace yields a flat solution when a straight line is present. Secondly, omitting $\mathbf{D}^{-1}$ is prone to nearly-flat solutions. These solutions place all curves, except a small number, near a single plane. Almost all intersection points

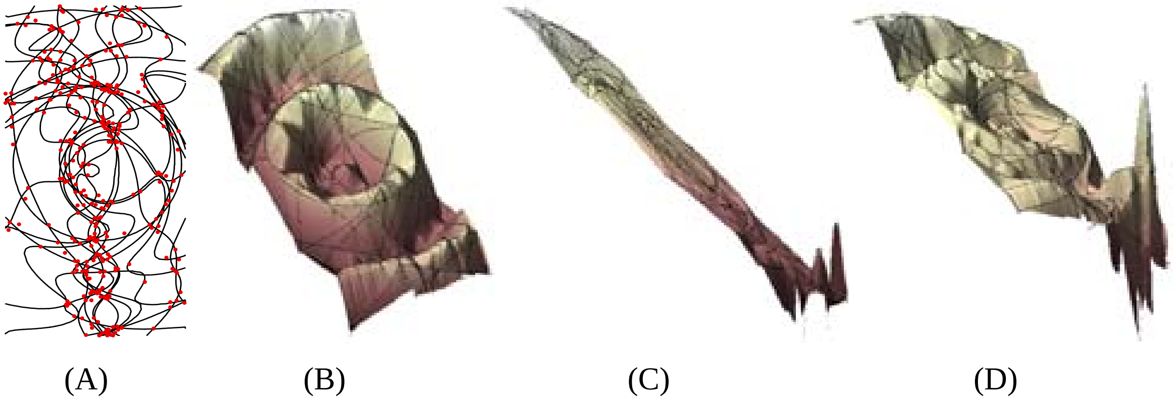


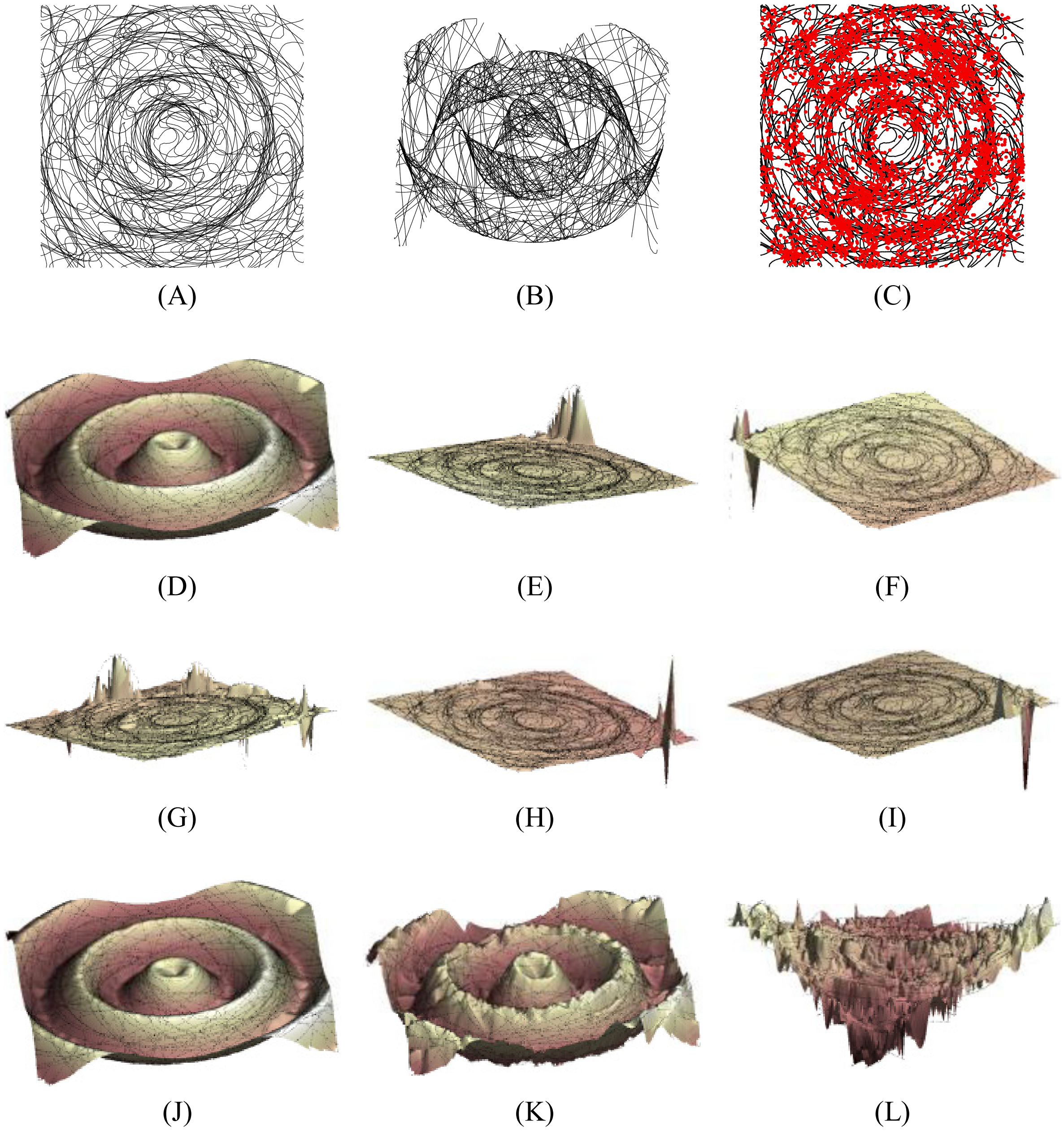
Figure 3.9: (A) 25 synthetic cross sections of a radial sine. The 351 intersection points are perturbed (red dots). (B) Our solution. (C) Last singular vector of the simple SVD method. This is a nearly-flat solution (note the bottom-right curve). (D) Second-last singular vector of the simple method. Although resembling the solution, it must be perpendicular to the last singular vector (C).

contribute nearly zero to the total error, and the error at the small number of intersection points of curves far from this plane might be relatively negligible. Note also that unlike the condition $\|\mathbf{Cv}\| = 1$, the condition $\|\mathbf{v}\| = 1$ has no geometric meaning. In fact, it mixes units of slope (a_i, b_i) and depth (d_i). Thus, changing the measurements units can potentially alter the singular values of the matrix (although the effect might be unnoticeable if the error in the system is small).

Figures 3.9 and 3.10 compare both methods on synthetic randomly oriented cross sections of a radial sine. The positions of the intersection points were perturbed and shown by red dots. Both methods succeed solving these examples without the perturbations. However, the perturbations were enough to break down the simple SVD method while our method succeeded. Figure 3.11 demonstrates our method when straight curves are present.

3.3.6 Ways to improve accuracy

While the method we presented could be good enough for visualization purposes, for other applications, notably 3D scanning by structured light, it might be desired to obtain the highest possible accuracy. In this section we outline several ways to improve the accuracy of our



(A) (B) (C)

(D) (E) (F)

(G) (H) (I)

(J) (K) (L)

Figure 3.10: (A) 100 synthetic cross sections of a radial sine. These curves intersect at 4475 points. (B) A second view of the curves overlaid on the 3D surface. (C) The red dots denote a random perturbation of the intersection points. (D) The output of the simple SVD method when run on the noiseless input (A). (E)-(I) The last five singular vectors produced by the simple SVD method when run on the noisy input (C). (J) The output of our method when run on the noiseless input (A). (K)-(L) The last two singular vectors produced by our method when run on the noisy input (C).

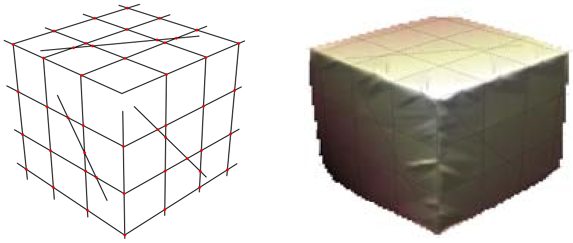


Figure 3.11: Left: adding four straight segments to figure 3.1 resolved the ambiguity. Right: computed surface by our method. The Last four singular vectors of the simple SVD method are flat, and the shape appears only at the fifth (note that our method thresholds by ε the singular vectors of $\mathbf{C}$, not $\mathbf{A}$).

method. We did not experiment with these enhancements since our focus was on the mathematical formulation. These practical enhancements should be considered for real implementations depending on the level of control one has on the scanning environment.

Our experience with our method has been that it is more stable than the simple linear method, in the sense that we never observed a case where it broke down and the simple linear method didn't, but observed the other way around (see figure 3.9). However, sometimes both of them break down. The common cause for breakdowns are curves having nearly-linear intersection points in the image which are not linear in 3D. The simplest solution is to drop these curves. However, if the curves are linear in 3D dropping them means throwing away some information.

Another way to improve the stability and accuracy is to surround the object with planar plates at various orientations, as illustrated in figure 3.12. This will help to prevent the image curves from being straight lines. An additional advantage of such a setup is that the image intersection points of curves on these planar plates can be computed more accurately. Since the image curves on planar plates are straight lines, one can do line fitting and intersect these lines with subpixel accuracy.

It is also possible to improve the numerical stability of the system by using restricted sets of planes. For instance, Bouguet et al. [14] used planes that intersect at a point. Caspi and Werman [21] used pencils of planes that have a common line. Sagawa et al. [165] used two

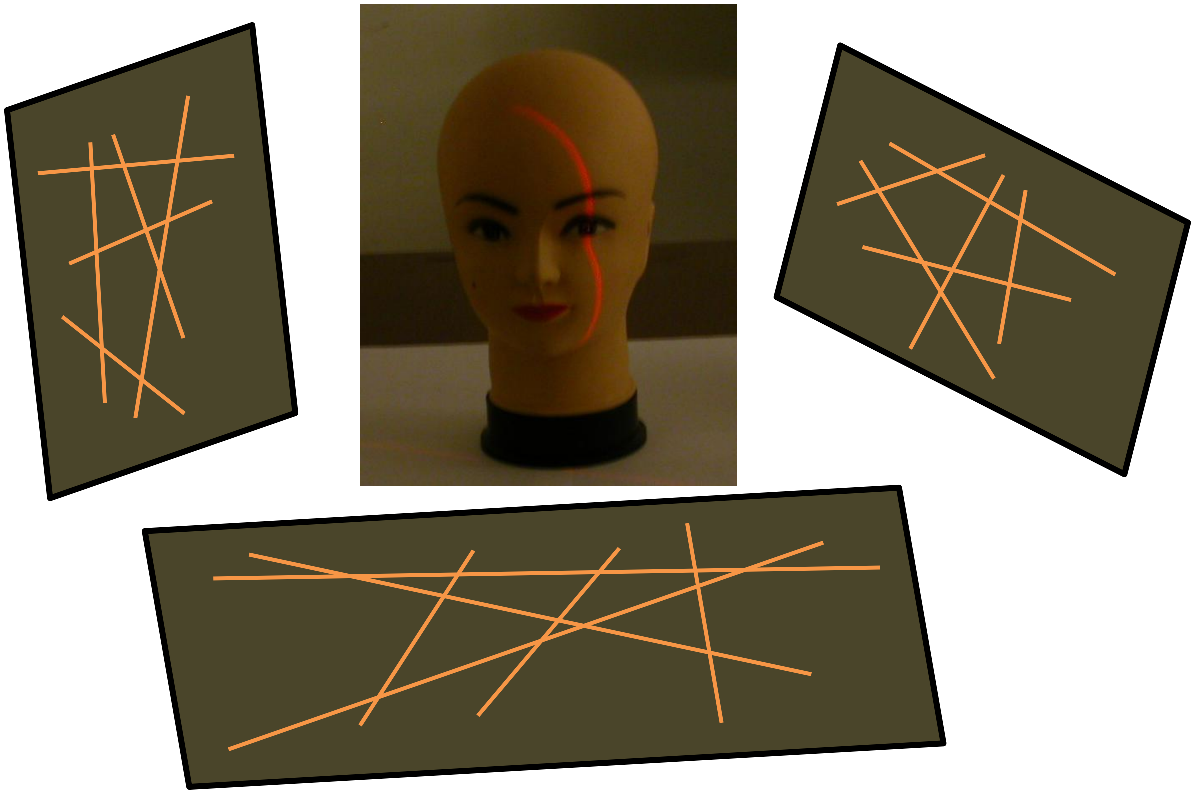


Figure 3.12: Surrounding the object with planes can improve the localization of intersection points.

pencils (horizontal and vertical strips) and report improved stability compared to systems of arbitrary planes. It should be kept in mind that due to possible concavities on the surface, using a single pencil of planes might not be enough to illuminate all the areas on the surface that are visible to the camera. However, it is possible to use multiple pencils while still keeping the number of variables small.

A different error model for noisy systems is to assume that the observed intersection points in the image are contaminated by i.i.d. isotropic Gaussian noise. The maximum likelihood estimation is achieved by minimizing the sum of squared distances between the observed intersection points (x_{ij}, y_{ij}) to the image-projection of the intersection lines L_{ij} (2.3) of the estimated planes (see figure 3.13). For orthographic projection the expression is

$$\sum_{ij} \frac{((a_i - a_j)x_{ij} + (b_i - b_j)y_{ij} + (d_i - d_j))^2}{(a_i - a_j)^2 + (b_i - b_j)^2} \text{ s.t. } \mathbf{v} \perp \mathrm{Null}(\mathbf{C}) \ , \tag{3.12}$$

and for perspective projection the expression is

$$\sum_{ij} \frac{((a_i - a_j)x_{ij}/f + (b_i - b_j)y_{ij}/f + (c_i - c_j))^2}{(a_i - a_j)^2 + (b_i - b_j)^2} \text{ s.t. } \mathbf{v} \perp \mathrm{Null}(\mathbf{C}_f) \ . \tag{3.13}$$

In (3.12), the numerators are the same as (2.2). The fractions are homogenous in the variables vector $\mathbf{v}$ (scale invariant). Note that (3.12) is invariant to addition of basic trivial component vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, which on their own yield the undefined value $0/0$. Optimization along the lines of (3.12) was proposed by Kawasaki and Furukawa [91] without the condition $\mathbf{v} \perp \mathrm{Null}(\mathbf{C})$. Without this condition, the optimization procedure is free to add unbounded trivial components, and converge towards $0/0$ which is numerically undesirable. There is a large variety of numerical methods that can be applied to (3.12), some of them are specifically designed to minimize homogeneous rational functions [29, 86]. These numerical methods typically converge to a local minimum, depending on the initial guess. Fortunately, the linear method usually provides a good starting point for nonlinear optimization.

When the system of curves is under-determined, or when a metric reconstruction is re-

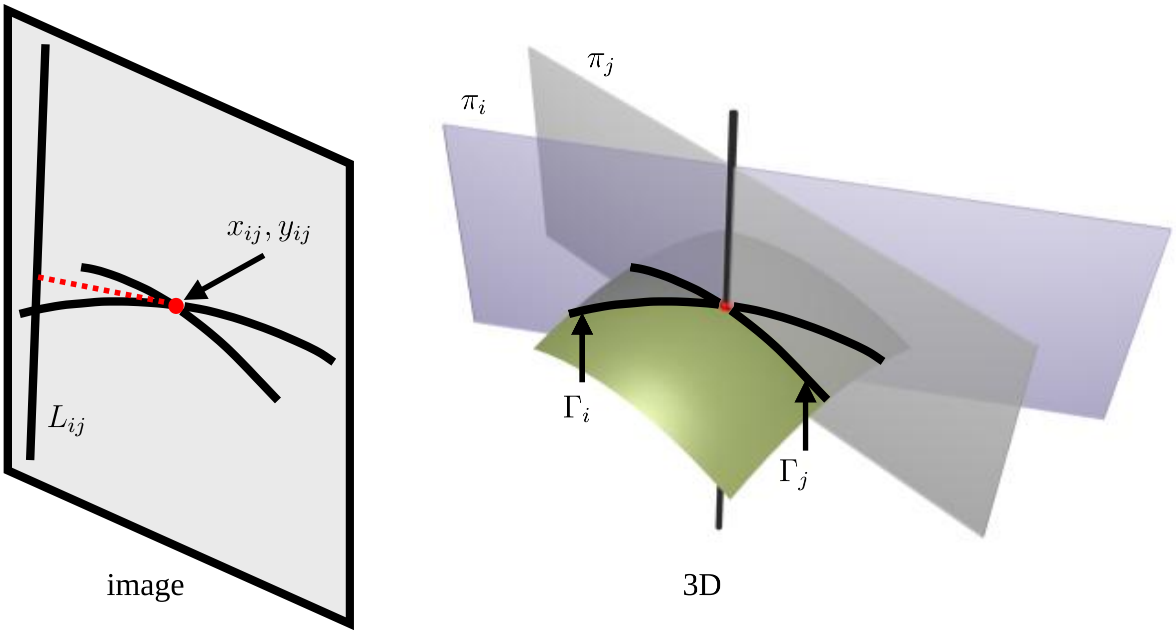


Figure 3.13: In a noisy system, an observed intersection point may not lie on the projection of the intersection line of the two 3D planes. We would like to minimize this distance.

quired, planarity constraints can be combined with additional depth cues. For example, Lipson and Shpitalni [119, 198] maximized measures such as corner orthogonality and verticality of line segments. Another example is Surfaces Of Revolution (SOR [31]), which are SHGC's with two additional constraints. Shimshoni and Ponce [173] and Shimodaira [172] used also shading. These works require nonlinear optimization, potentially in high dimension. In contrast, Sugihara [183] already proposed to simplify the search by first solving the linear constraints to obtain a low-dimensional linear subspace that contains the solution, and then searching for scale and trivial subspace coefficients (e.g. search in a 4D space) to optimize additional objectives. The benefit of this approach is that the non-linear search is confined to a very low dimension. If we ignore the global depth ambiguity ($\mathbf{v}_3$), the 3D space of GBR ambiguities can even be sampled densely enough to get close to the solution of nonlinear cost functions (a parallel brute-force search might be the method of choice for the brain). In practice, it is sometimes helpful to expand the search space with additional singular vectors of $\mathbf{C}$ that correspond to small singular values. Optimization in a subspace has been demonstrated for several nonlinear cost functions. Sugihara applied this idea to shape from shading and shape from texture of over-constrained polyhedra. Liu et al. [121] used nonlinear search in a low dimension to optimize the minimum standard deviation of angles (MSDA). Kawasaki and Furukawa [89] used the low-dimensional space to optimize orthogonality constraints in structured light. Since our method captures the trivial subspace in a more geometrically meaningful way than previous methods, it can contribute in that direction as well.

3.4 Parallel planar curves

In the previous sections we discussed systems of planar curves with many intersection points. Next we look at the complimentary case of parallel cross sections, where standard shape from texture algorithms cannot be applied. Consider the topographic maps in figure 3.14. In this problem one has to set the order and distances (in z) between the planes. Notice the strong depth perception in absence of elevation values.

It seems that in general humans are better at perceiving over-constrained polyhedra than arbitrarily oriented planar cross sections (e.g. figure 3.9 (A)). Although the problem of interpreting parallel planar curves is very under-constrained, there are examples where humans have strong 3D perception. Todd and Reichel [188] showed an example where humans perceive parallel cross sections better than randomly oriented cross sections with plenty of intersection points. Whether humans are able to perceive a 3D shape from its parallel cross sections varies with the shape and the orientation of the cross sections. In particular, topographic maps will not always pop-out naturally to the correct shape of the terrain. The same shape sliced at different orientations can generate different percepts (or the perception of curves on a flat surface). Therefore, we are not particularly interested here in Euclidian reconstruction of topographic maps with fixed depth gaps between adjacent curves, but in suggesting plausible smooth shapes using a mechanism similar to the case of intersecting planar curves.

To reduce the parallel case to the case of intersecting curves we use a simple shape prior,

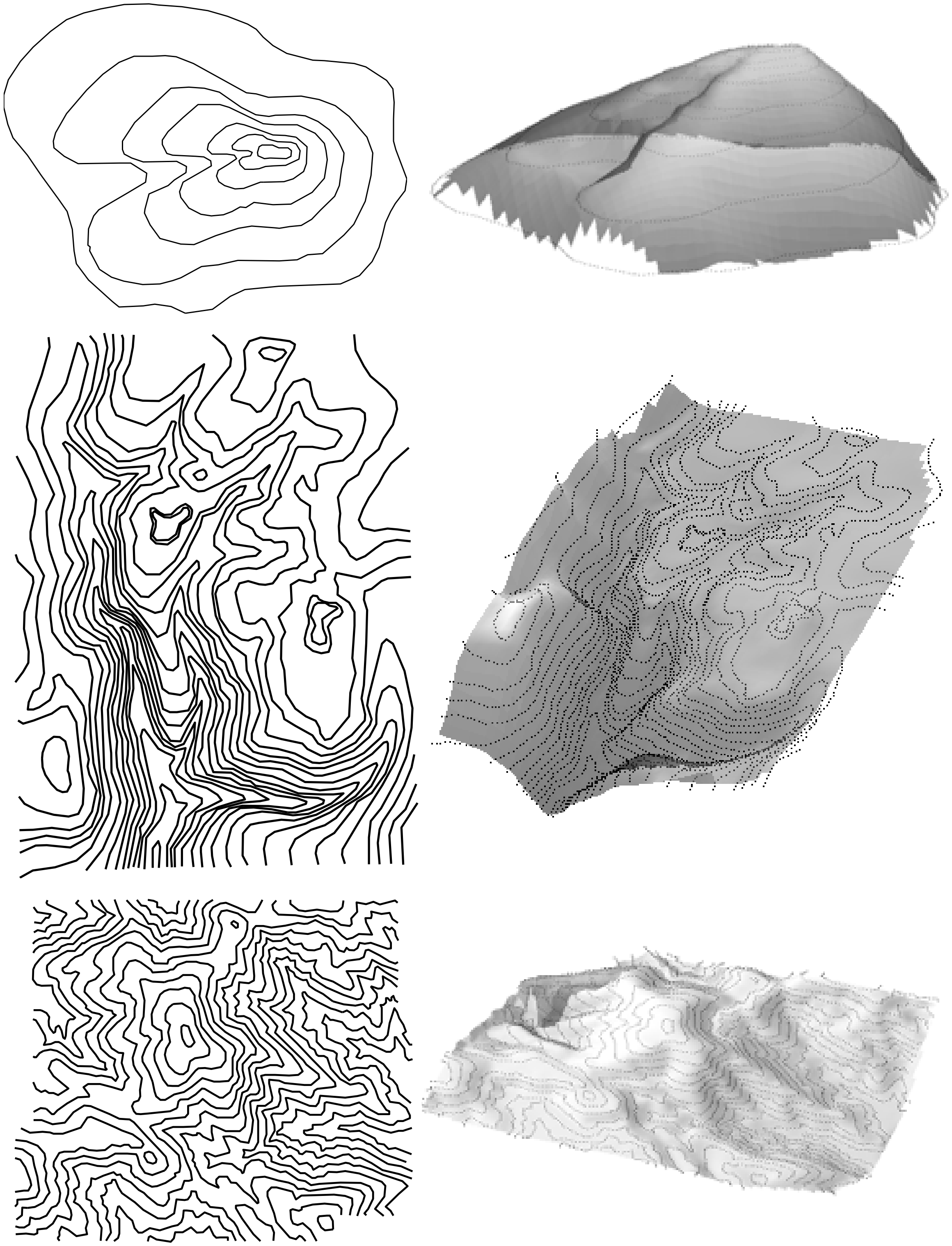


Figure 3.14: 3D shape estimation from parallel planar curves (topographic maps).

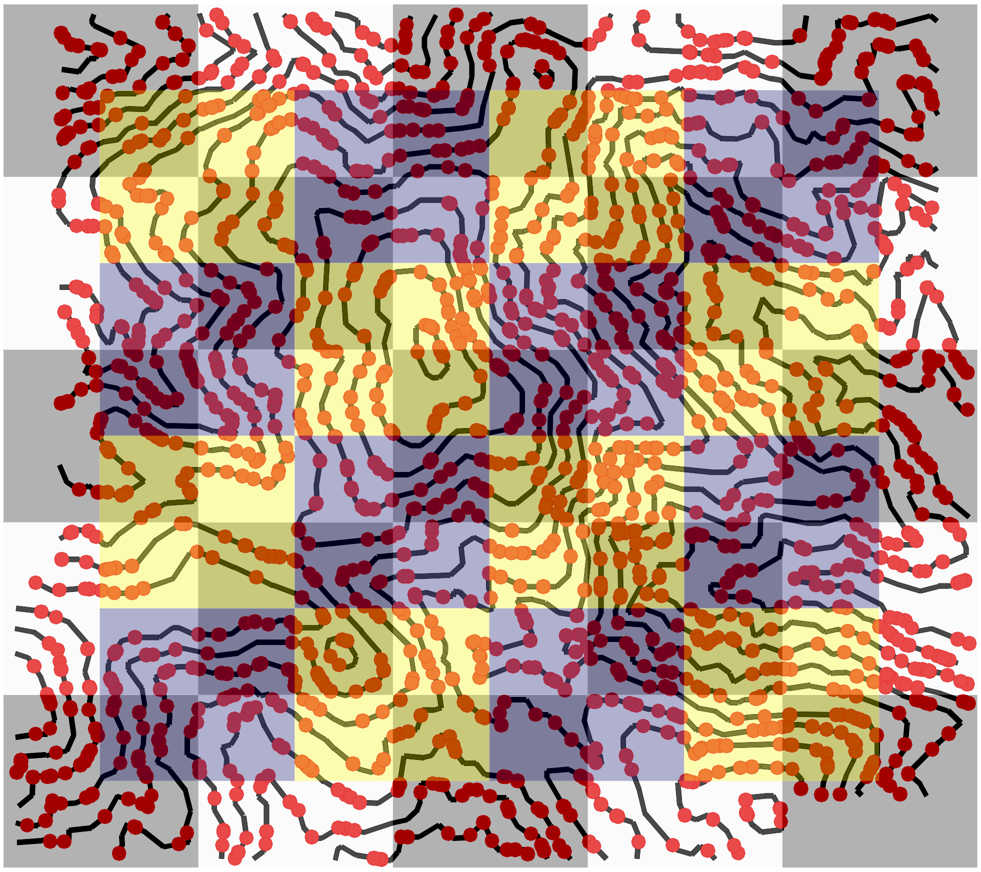


Figure 3.15: Reconstruction of topographic maps by piecewise planarity using a grid of 10×10 cells. Each window is made of 2×2 cells. The sampled red dots are considered as approximate intersection points between the planar cross sections to the planar cell windows.

namely assume that the surface is roughly planar over local patches. The construction is shown in figure 3.15. We overlay a coarse rectangular grid over the image region, and group neighboring 2×2 cells into a set of W partially overlapping windows. The windows serve as virtual facets. We sample a set of points (x_i, y_i) along the curves. These points are treated as the image of the intersection points between the planar curves and the virtual planar facets.

Although in general the planes will not intersect at the sampled points, we may still strive to minimize the distances. Rather than parameterizing the facets explicitly, for each window we measure the distances between the curves to the best fitting plane at the sampled points. This is done by constructing a matrix $\mathbf{C}_i$ as in (3.1) for each window (normalized by the number of points in each window). To enforce the curves to be parallel, we can either set $a_i = b_i = 0$ and solve for $\mathbf{v} = (d_1, \ldots, d_N)^T$, or alternatively add a penalty for the variance

$$\|\mathbf{R}\mathbf{v}\|^2 = \lambda \left(\mathit{Var}(a_i) + \mathit{Var}(b_i) \right) \ , \tag{3.14}$$

$$\mathbf{R} = \sqrt{\frac{\lambda}{N}} \begin{bmatrix} \mathbf{I}_N - \frac{1}{N}\mathbf{1}_{N\times N} & \mathbf{0}_{N\times N} & \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} & \mathbf{I}_N - \frac{1}{N}\mathbf{1}_{N\times N} & \mathbf{0}_{N\times N} \end{bmatrix} .$$

The choice of λ depends on our confidence that the planes are parallel. The optimization problems becomes (semicolons denote vertical matrix concatenation):

$$\underset{\mathbf{v}}{\operatorname{argmin}} \left\| \left[\mathbf{R}; \tfrac{1}{\sqrt{W}}\mathbf{C}_1; \cdots ; \tfrac{1}{\sqrt{W}}\mathbf{C}_W \right] \mathbf{v} \right\| \ \text{s.t.} \ \|\mathbf{C}\mathbf{v}\| = 1 \ . \tag{3.15}$$

This problem is solved as in (3.6). As before, the solution is up to GBR ambiguity. Examples are shown in figure 3.14, where we used a grid of 10×10 cells, and each window is made of 2×2 cells.

The method proposed has several limitations. First, it depends on the size of the windows that roughly approximate the surface. When they are too small the surface can be over-smoothed, as shown in figure 3.16. Although these shapes do not seem as smooth as the true surface, they are actually smoother according to our measure.

Another technical limitation of our formulation is that we cannot group all sampled points to the same plane, since minimizing $\|\mathbf{C}\mathbf{v}\|$ s.t. $\|\mathbf{C}\mathbf{v}\| = 1$ is meaningless. It is interesting to note that several unresolved examples in the literature could be solved by arranging the cross sections as close as possible to a single principle plane while keeping the shape non-flat. Computationally this means finding the singular vector of the matrix $\left[\mathbf{R}; \mathbf{C}\right]$ that corresponds to the smallest singular value greater than ε. Figure 3.17 demonstrates this singular vector on examples from Stevens [181], Todd and Reichel [188], and two synthetic examples with random gaps between parallel planes and nearly-parallel planes.

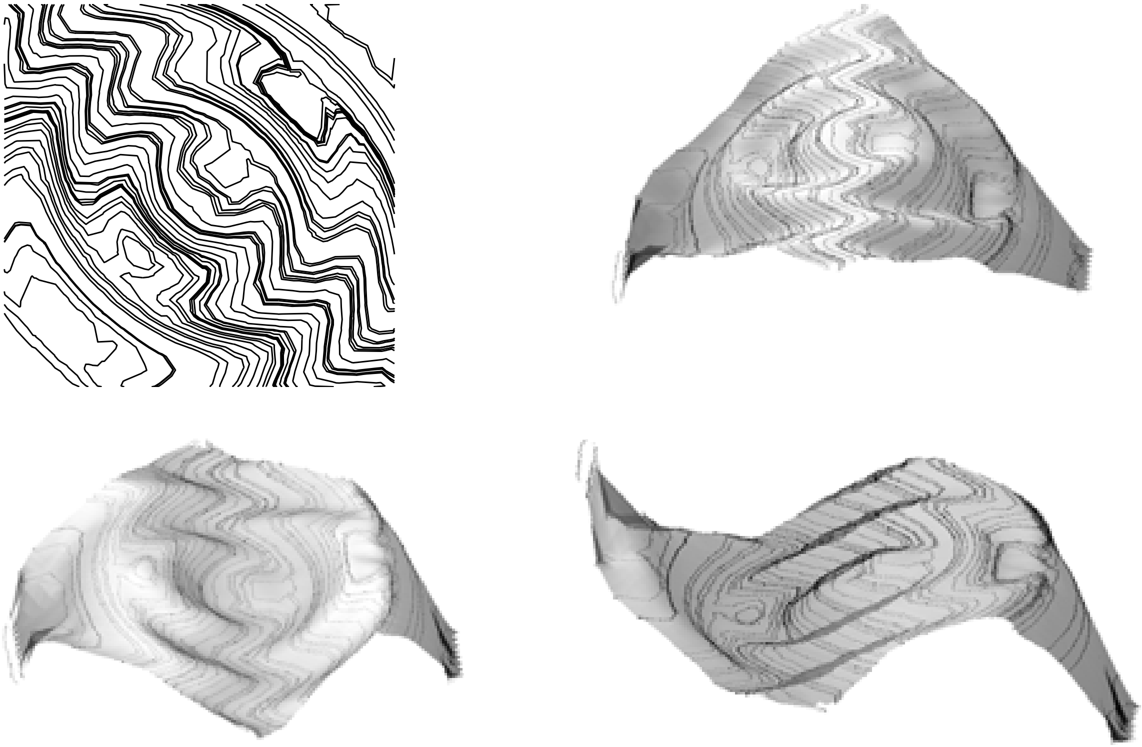


Figure 3.16: Clockwise: synthetic cross sections of a radial sine and last three singular vectors, solved by piecewise planarity on a grid of 10×10 cells. These solutions over-smooth the surface where a single plane fit (figure 3.17(C)) is appropriate for this shape.

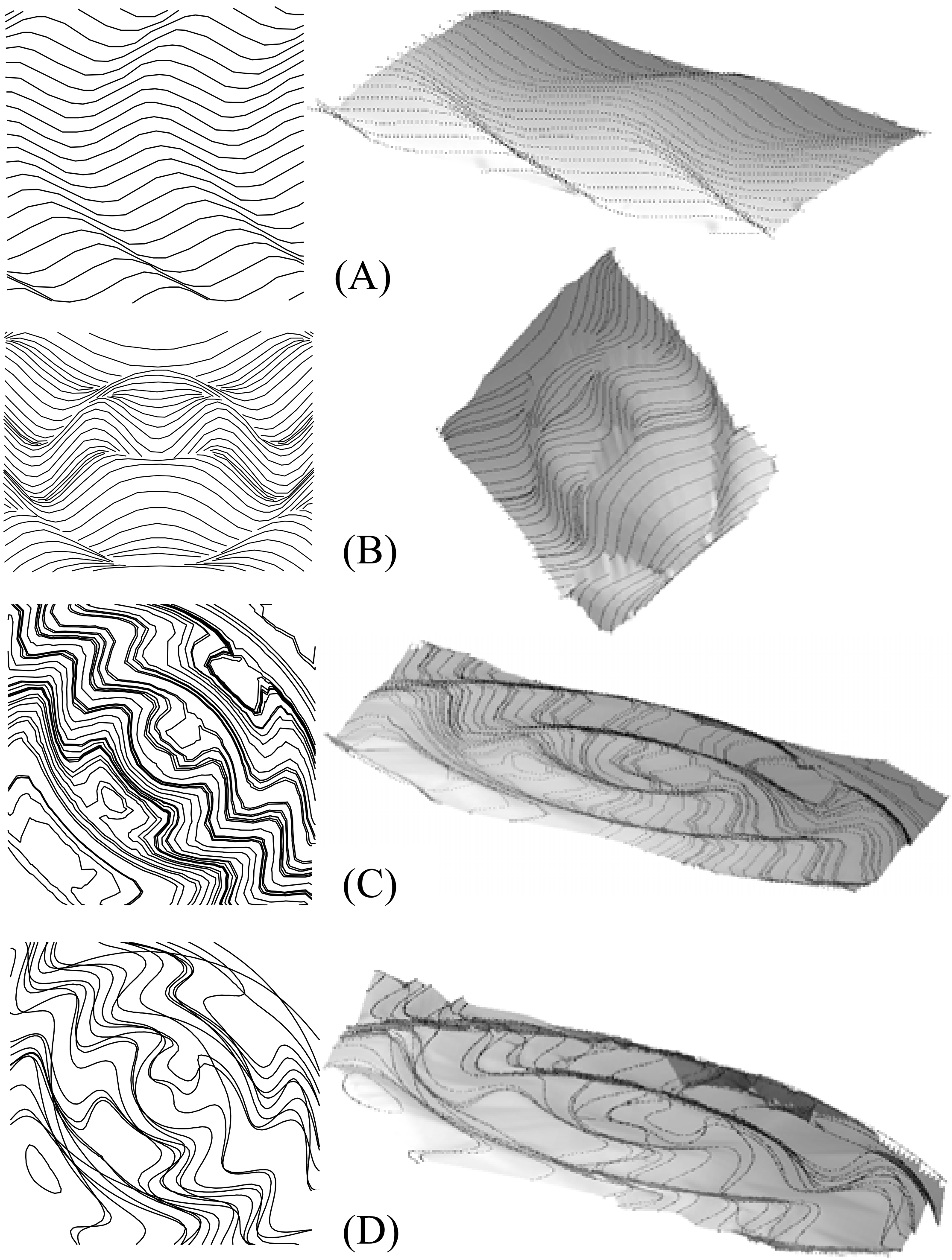


Figure 3.17: Shapes determined by a single plane fit. (A) Waves from Stevens [181]. (B) Radial sine from Todd and Reichel [188]. (C) Sine with non-uniform cuts (same as figure 3.16). (D) Sine with perturbed parallel cuts (intersection points were not used). $\lambda = 10^4$ for (A,B,C) and $\lambda = 10^3$ for (D).

3.5 Conclusions

In this chapter we presented a unified analysis of shape from planar curves. We generalized the linear method of Bouguet et al. to deal with arbitrary planes, improved its robustness, and demonstrated its applicability to single view modeling and fully uncalibrated structured light. As the cases of polyhedra, SHGC's, ZGC's and uncalibrated structured light were treated independently in the literature, it is valuable that the same approach can deal with all of them.

A difficult issue we weren't dealing with is assigning curves to planar faces automatically, which is a perceptual organization and grouping problem. One may try heuristics such as grouping parallel lines to a planar face, yet often multiple interpretations are possible.

We also experimented with parallel planar curves where we assumed the curves intersect implicit planar facets. While the case of intersecting planar curves is now well understood, the case of parallel cross sections is more challenging and may require the combination of additional depth cues or prior knowledge. The method we proposed can be used to extract a low dimensional subspace in which additional non-linear search can be performed.

Chapter 4

SDP Heuristics for Surface Reconstruction Ambiguities

4.1 Introduction

An important problem in surface reconstruction is the handling of situations in which there are not enough constraints to uniquely determine the surface shape. In these under-constrained situations there are multiple interpretations of the surface that are consistent with the available constraints. The ambiguities can be continuous, such as unknown depth, or discrete, such as in/out reversal. In this chapter we deal with constraints that have discrete ambiguities. Such constraints are common to many surface reconstruction problems. They arise in areas such as shape from texture, shape from shading, photometric stereo, shape from defocus, and local structure from motion [194].

Our main interest is resolving the inherent ambiguities in shape from texture. Textured surfaces, such as the vase shown in figure 4.1, produce a strong 3D perception. Surface normals can be estimated from the texture up to two-fold ambiguity. The interpretation of such images poses two interconnected challenges. First, one has to choose which of the possible normals is the correct one. Second, one has to integrate the normals to a plausible surface, which

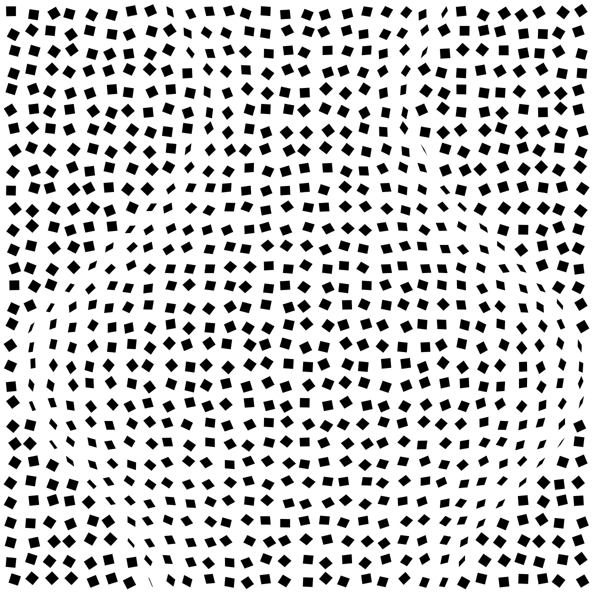


Figure 4.1: A synthetically textured surface.

preferably should be smooth. Note that it is important that surface smoothness is taken into account in between the texture elements, as explained in figure 4.2.

The problem of smooth surface integration under ambiguous constraints is computationally hard. In particular, it does not meet the conditions needed for graph-cut algorithms to produce optimal solutions [25]. We show it can be approximated with semidefinite programming (SDP). Zhu and Shi [218] used SDP to solve in/out reversal ambiguities of surface patches in shape from shading. We show that a similar mathematical formulation applies to other surface reconstruction problems.

The general approach starts by representing the surface as a spline, i.e. the shape is controlled by a set of continuous variables. Additional discrete variables are used to form the ambiguous constraints. A quadratic cost function measuring surface smoothness and constraint satisfaction is defined. The continuous variables are eliminated, leading to a quadratic cost function in the discrete variables only. An SDP relaxation embeds the discrete variables in a continuous high dimensional space. Finally, a rounding step sets the discrete variables and proposes a 3D shape.

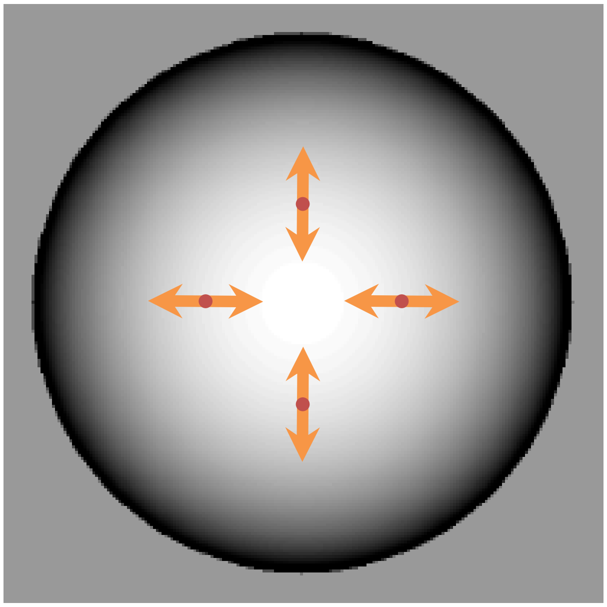


Figure 4.2: An example where it is important to integrate the smoothness over the entire surface. Consider a spherical surface (or more generally, an ellipsoidal surface) that can be convex or concave. Suppose we can infer the ambiguous normals from texture at the four positions of the circles shown above. These (non-unit) normals are of the form $(\pm a, 0, 1)^t$, $(0, \pm b, 1)^t$, and the norm of the difference between each normal to its nearest neighbor is $\|(\pm a, 0, 1)^t - (0, \pm b, 1)^t\| = \sqrt{a^2 + b^2}$. If our cost function only compares each normal to its two nearest-neighbor normals, then any assignment to the possible flips has the same score. Thus, for this specific configuration, relying on nearest neighbors of the normals is not helpful, and surface smoothness has to be taken into account in between the texture elements.

The problems we deal with are larger than those considered by Zhu and Shi, and the standard Goemans-Williamson random hyperplane rounding technique [59] will usually produce sub-optimal results. We describe several heuristics that may improve the quality of the solutions.

4.2 Problem formulation

We show below that several surface reconstruction problems can be written in the form

$$\underset{\mathbf{v},\mathbf{d}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{v}-\mathbf{B}\mathbf{d}\|^2 \ , \tag{4.1}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices, $\mathbf{d}{\in}\{-1,1\}^n$ is a vector of discrete decision variables, and $\mathbf{v}{\in}\mathbb{R}^m$ is a vector of continuous parameters that controls the surface. We represent the surface as a spline, i.e. a linear combination of basis functions

$$z(x,y)=\sum_{i=1}^{m}\mathbf{b}_i(x,y)\mathbf{v}_i \ . \tag{4.2}$$

The specific bases we use are described in Appendix A. Controlling the surface by a relatively low dimensional vector of parameters $\mathbf{v}$ reduces the computational load and prevents overfitting noisy constraints. Next we discuss several specific problems that can be formulated in this form. Algorithms for solving (4.1) will be discussed in section 4.3.

4.2.1 Shape from two-fold ambiguous normals

Traditional shape from texture deals with estimating surface normals from the distortion of texture under projection. Under orthographic projection, normal estimates usually have a two-fold tilt ambiguity, because the projections of local planar patches with normals $(p,q,1)$ and $(-p,-q,1)$ are identical. We address the problem of estimating the shape of a surface given a large number of ambiguous normal estimates, as demonstrated in figure 4.3. In this work we use the simplest way to recover the normals, which is from the distortion of known texture elements. Humans are probably relying on density changes to estimate the normals. In both ways the estimation of normals is up to two-fold ambiguity.

For sparse texture the problem is under-constrained, since the surface can be integrated for any choices of the normals. In addition, under orthographic projection there is one global in/out reversal ambiguity, and a continuous ambiguity in absolute depth. However, if we make the assumption that the surface is smooth, we can identify the more probable shapes of the surface. In related work, Forsyth [48, 49] proposed alternating between optimizing surface smoothness and selecting the normals. We show in section 4.3 that by using a quadratic smoothness term the problem can be converted into an entirely discrete optimization problem.

Denote the partial derivatives of the surface by $p = \frac{dz}{dx}$, $q = \frac{dz}{dy}$. The texture observed at a specific image point, say (x_i, y_i), provides two choices for the surface derivatives, namely $(p_i, q_i)\mathbf{d}_i$, with $\mathbf{d}_i = \pm 1$. By differentiating (4.2) with respect to x and y, we can express the derivatives of the spline surface in terms of the vector $\mathbf{v}$. This provides two known row vectors $\mathbf{a}_{p_i}$ and $\mathbf{a}_{q_i}$ such that

$$(0,\ldots,0,p_i,0,\ldots,0)\mathbf{d} = \mathbf{a}_{p_i}\cdot\mathbf{v} \quad , \quad (0,\ldots,0,q_i,0,\ldots,0)\mathbf{d} = \mathbf{a}_{q_i}\cdot\mathbf{v}\ , \tag{4.3}$$

where $\mathbf{d}$ is an n-vector formed from the sign bits $\mathbf{d}_i$.

To regularize the surface we use a quadratic smoothness term. The smoothness term can be expressed in terms of the spline parameters using a matrix $\mathbf{E}$ such that $||\mathbf{E}\mathbf{v}||^2$ is the smoothness energy (see Appendix A for more details). The smoothness energy is weighted with a regularization parameter λ that balances between the smoothness energy and the constraint error. Together, these terms can be written in the form of (4.1),

$$\operatorname*{argmin}_{\mathbf{v},\mathbf{d}} \left\| \underbrace{\begin{bmatrix} \sqrt{\lambda}\mathbf{E} \\ \mathbf{A}' \end{bmatrix}}_{\mathbf{A}} \mathbf{v} - \underbrace{\begin{bmatrix} 0 \\ \mathbf{B}' \end{bmatrix}}_{\mathbf{B}} \mathbf{d} \right\|^2 . \tag{4.4}$$

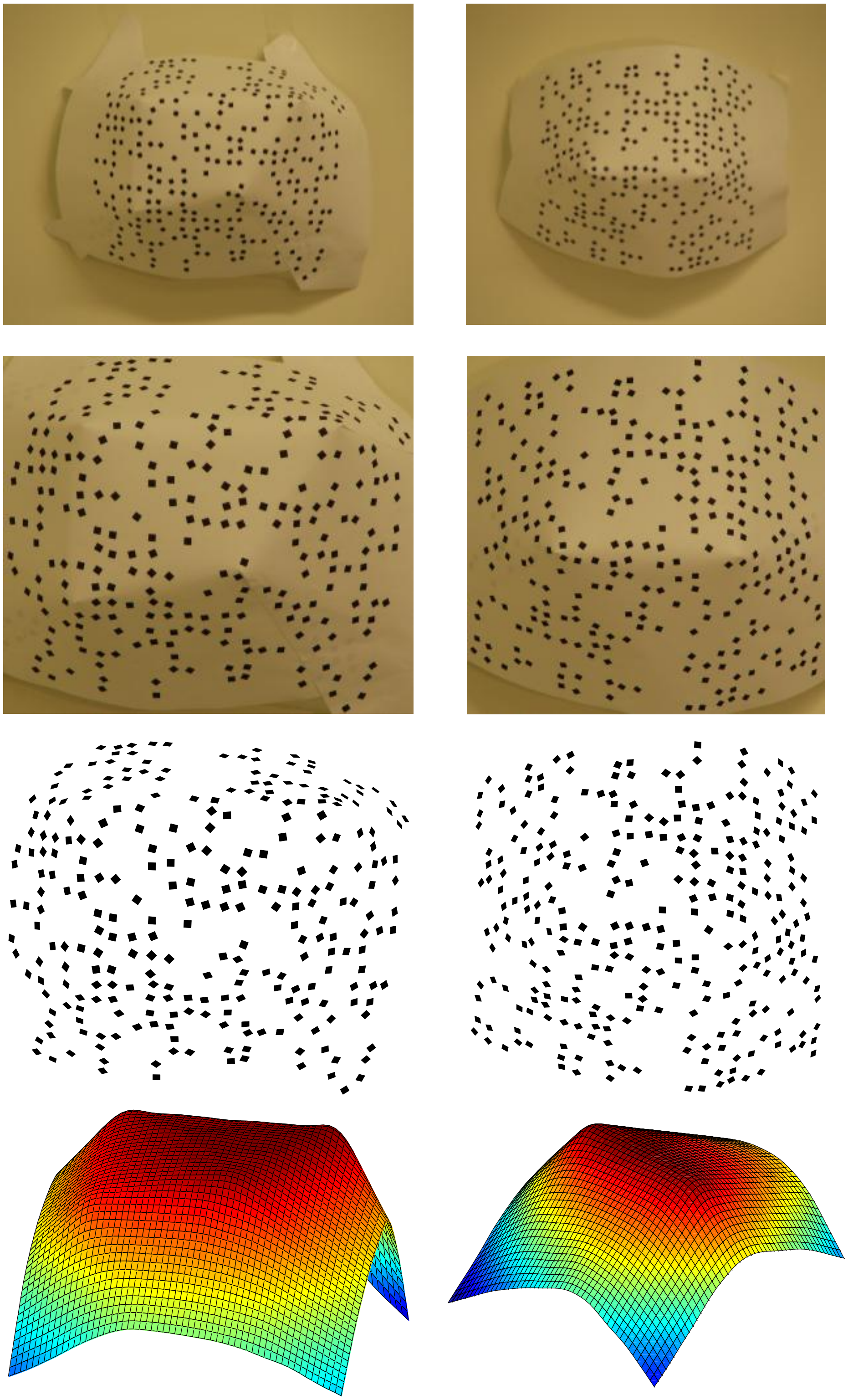


Figure 4.3: Top to bottom: surfaces textured with squares, input images, parallelograms extracted from the images and the computed surfaces.

Here $\mathbf{A}'$ and $\mathbf{B}'$ are matrices formed from the constraints in (4.3), with each constraint in (4.3) appearing as a single row in $\mathbf{A}'\mathbf{v} = \mathbf{B}'\mathbf{d}$.

4.2.2 Two-light photometric stereo

In standard photometric stereo of a Lambertian surface, at least three light sources at known positions are used to determine the surface normals. For a given light source, the image brightness at a point constrains the corresponding surface normal to a circle on the unit sphere. Two light sources may limit the normal to two possibilities, which are the intersections of the two circles. A third light is usually enough to disambiguate the normal. A two-fold ambiguity remains when a point is in shadow with respect to one of the lights [70], or in degenerate configurations, when all the light directions are located on a great circle on the unit sphere. Under a crude approximation, this happens with the sun as the light source during the daily rotation of the earth (assuming the earth's axis of rotation is perpendicular to the direction to the sun. This assumption is inaccurate).

In special cases, two lights are sufficient. This occurs when the two circles on the unit sphere touch at a point, or when one of the intersection points of the two circles is on the occluded hemisphere. Onn and Bruckstein [141] studied photometric stereo of Lambertian surfaces using two lights. Their method uses the points that are uniquely determined by two lights to divide the image into regions. Inside each region integrability is used to choose between the two possibilities of the normal. However, detecting the boundaries of these regions on a discrete grid is susceptible to errors, especially when the surface has discontinuous derivatives.

Our formulation for two-light photometric stereo avoids region detection and adds a surface smoothness prior. We assume knowledge of points (x, y) where we have two choices for the surface derivatives, (p_1, q_1) or (p_2, q_2). Onn and Bruckstein [141] derived the formulas for the possible derivatives in the Lambertian case. For other shading models these choices may be determined experimentally. However, more complicated shading models might require more images, since the corresponding two curves on the unit sphere might intersect more than twice.

We don't deal with these cases here.

The two choices for the surface derivatives at point (x, y) can be expressed as functions of a sign bit $\mathbf{d}_{xy} = \pm 1$

$$\begin{aligned} \begin{pmatrix} p \\ q \end{pmatrix} &= \begin{pmatrix} p_{sum} \\ q_{sum} \end{pmatrix} + \begin{pmatrix} p_{\mathit{diff}} \\ q_{\mathit{diff}} \end{pmatrix} \mathbf{d}_{xy} \quad , \\ \begin{pmatrix} p_{sum} \\ q_{sum} \end{pmatrix} &= \frac{1}{2} \begin{pmatrix} p_1 + p_2 \\ q_1 + q_2 \end{pmatrix} \quad , \quad \begin{pmatrix} p_{\mathit{diff}} \\ q_{\mathit{diff}} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} p_1 - p_2 \\ q_1 - q_2 \end{pmatrix} \quad . \end{aligned} \tag{4.5}$$

As in the previous subsection, the spline derivatives can be written as $p = \mathbf{a}_p \cdot \mathbf{v}$, $q = \mathbf{a}_q \cdot \mathbf{v}$. Collecting the equations for all points and adding the smoothness term we arrive at

$$\operatorname*{argmin}_{\mathbf{v},\mathbf{d}'} \left\| \begin{bmatrix} \sqrt{\lambda}\mathbf{E} \\ \mathbf{A}' \end{bmatrix} \mathbf{v} - \begin{bmatrix} 0 \\ \mathbf{B}' \end{bmatrix} \mathbf{d}' - \begin{pmatrix} 0 \\ \mathbf{b}' \end{pmatrix} \right\|^2 = \operatorname*{argmin}_{\mathbf{v},\mathbf{d}'} \left\| \begin{bmatrix} \sqrt{\lambda}\mathbf{E} \\ \mathbf{A}' \end{bmatrix} \mathbf{v} - \begin{bmatrix} 0 & 0 \\ \mathbf{B}' & \mathbf{b}' \end{bmatrix} \begin{pmatrix} \mathbf{d}' \\ 1 \end{pmatrix} \right\|^2 . \tag{4.6}$$

Here the nonzero entries of the matrix $\mathbf{B}'$ are made of $p_{\mathit{diff}}, q_{\mathit{diff}}$, and the vector $\mathbf{b}'$ is made of p_{sum}, q_{sum} according to (4.5). A standard transformation to bring (4.6) to the form of (4.1) is to solve for $\mathbf{d} = \begin{pmatrix} \mathbf{d}' \\ 1 \end{pmatrix}$. Note that the cost of a pair $(\mathbf{v}, \mathbf{d})$ in (4.1) equals the cost of $(-\mathbf{v}, -\mathbf{d})$. Thus, if after solving (4.6) the last coordinate of $\mathbf{d}$ is -1, we need to negate the solution.

4.2.3 Segments of known length

Assume a collection of segments of known 3D length (or pairs of features with known 3D distances) is detected on a smooth surface, as shown in figure 4.4. Given an orthographic view, each segment can have a front/back reversal. Similar problems were considered by Naito and Rosenfeld [133] and Koenderink and van Doorn [104].

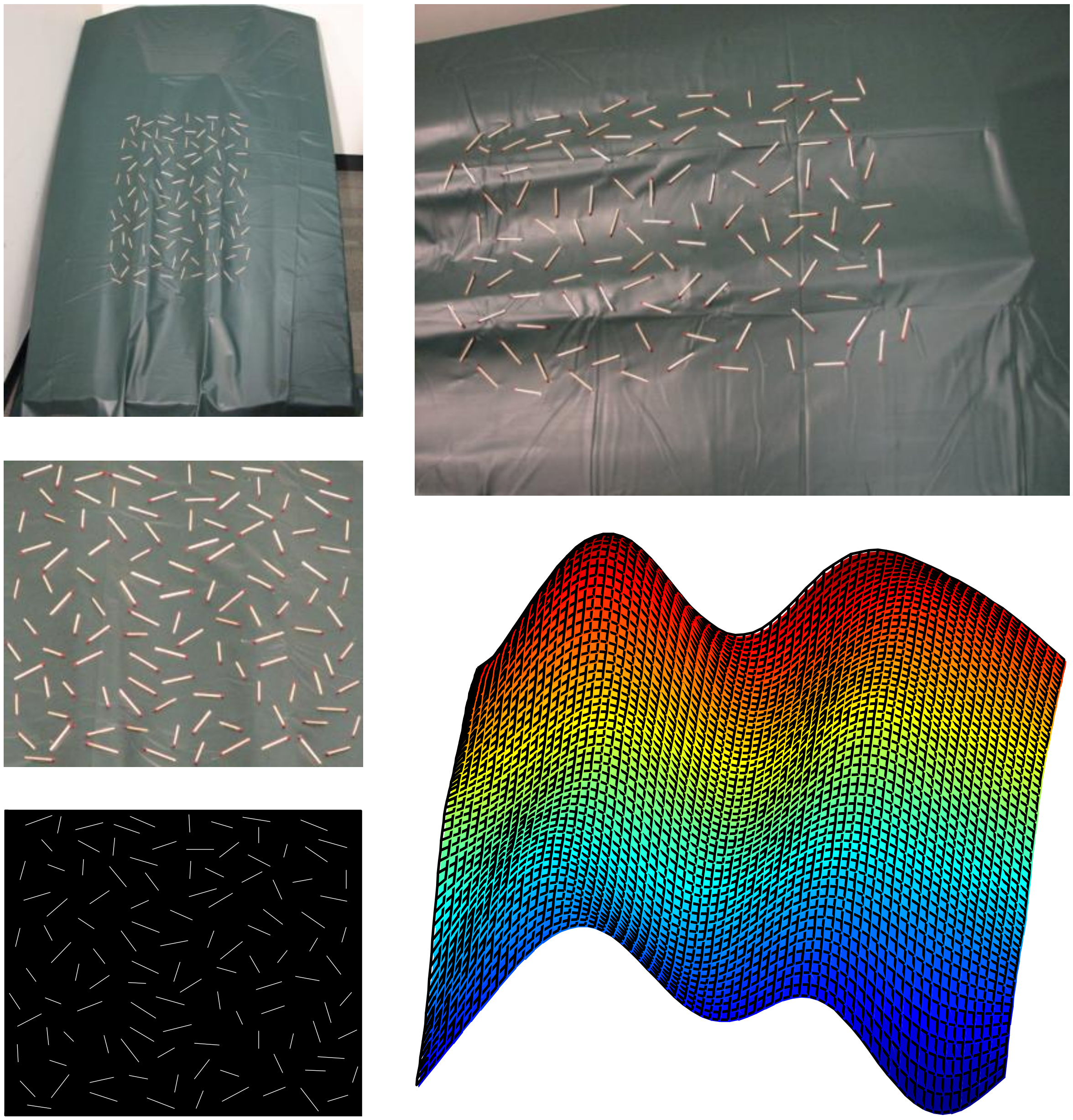


Figure 4.4: Top row: setup of 112 matchsticks on a surface. Middle left: the input image. Bottom left: line segments that were extracted from the input image. Bottom right: the computed surface using a spline with 50 basis functions.

The depth difference of the segment's endpoints is constrained by

$$z_i - z_j = \mathbf{d}_{ij}\sqrt{l^2 - r_{ij}^2} = \Delta_{ij}\mathbf{d}_{ij} \quad , \tag{4.7}$$

where $\mathbf{d}_{ij} = \pm 1$, l is the 3D length of the segment and r_{ij} is the observed length in the image. l, r_{ij} (and hence Δ_{ij}) are assumed to be known. By (4.2), the depth z_i at a point (x_i, y_i) is a linear combination of the spline bases at that point. That is, z_i can be expressed as $\mathbf{a}_i \cdot \mathbf{v}$, where $\mathbf{a}_i$ is a known row vector, and similarly for z_j. Each depth constraint can be written as

$$(\mathbf{a}_i - \mathbf{a}_j)\mathbf{v} = (0, \ldots, 0, \Delta_{ij}, 0, \ldots, 0)\mathbf{d} \; . \tag{4.8}$$

Collecting these equations over all constraints and adding the smoothness term can be written as in (4.4).

4.3 SDP rounding heuristics

We now turn into solving problems of the form (4.1). For any vector $\mathbf{d}$, the optimal $\mathbf{v}$ is

$$\mathbf{v} = \mathbf{A}^{+}\mathbf{B}\mathbf{d} \quad , \tag{4.9}$$

where $\mathbf{A}^{+}$ is the pseudo-inverse of $\mathbf{A}$. Plugging $\mathbf{v}$ back into equation (4.1) we get $\|(\mathbf{A}\mathbf{A}^{+}\mathbf{B} - \mathbf{B})\mathbf{d}\|^2 = \mathbf{C} \bullet \mathbf{X}$, where $\mathbf{C} = (\mathbf{A}\mathbf{A}^{+}\mathbf{B} - \mathbf{B})^t(\mathbf{A}\mathbf{A}^{+}\mathbf{B} - \mathbf{B}) = \mathbf{B}^t(\mathbf{I} - \mathbf{A}\mathbf{A}^{+})\mathbf{B}$, $\mathbf{X} = \mathbf{d}\mathbf{d}^t$, and $\bullet$ is the inner product of matrices $(\mathbf{C} \bullet \mathbf{X} = \sum \mathbf{C}_{ij}\mathbf{X}_{ij})$. Therefore, the problem is reduced into a combinatorial optimization problem of finding the discrete vector $\mathbf{d} \in \{-1, 1\}^n$ which minimizes $\mathbf{C} \bullet (\mathbf{d}\mathbf{d}^t)$. Once $\mathbf{d}$ is found, $\mathbf{v}$ is given by (4.9) and the 3D shape is given by (4.2).

Unfortunately, the general problem is NP-hard and difficult to approximate [2]. Semidefinite programming is widely used to find approximate solutions for problems of this kind. The standard SDP relaxation requires the matrix $\mathbf{X}$ to be symmetric positive semidefinite (instead of rank one) with ones on the main diagonal (since $\mathbf{d}_i^2 = 1$), i.e. solving

$$\underset{\mathbf{X}}{\operatorname{argmin}} \ \mathbf{C} \bullet \mathbf{X} \quad \text{s.t.} \quad \mathbf{X}_{ii} = 1 \ , \ \mathbf{X} \succeq 0 \ . \tag{4.10}$$

The SDP problem (4.10) is convex and can be solved in polynomial time using an SDP solver. A discrete vector $\mathbf{d}$ is obtained from the continuous solution matrix $\mathbf{X}$ in a rounding phase. The Goemans-Williamson random hyperplane rounding scheme [59] uses the Cholesky factorization of the matrix $\mathbf{X}$, $\mathbf{X} = \mathbf{R}\mathbf{R}^t$. Let $\mathbf{u}_i$ denote the i-th row of $\mathbf{R}$. Since $\mathbf{X}_{ii} = \mathbf{u}_i \cdot \mathbf{u}_i^t = 1$, the rows of $\mathbf{R}$ can be viewed as an embedding of the decision variables into the unit sphere in $\mathbb{R}^n$ (this embedding is not unique). Rounding is done by picking a random hyperplane with normal $\mathbf{N}$ and setting $\mathbf{d}_i = \operatorname{sign}(\mathbf{u}_i \cdot \mathbf{N})$.

For matrices $\mathbf{C}$ arising from the max-cut problem with nonnegative edge weights, this scheme provides a strong, provable expected approximation ratio of at least 0.878. Moreover, several derandomization schemes achieve this approximation ratio with deterministic algorithms [10]. However, this approximation ratio does not apply to our problem for several reasons. First, their analysis in deriving the 0.878 approximation ratio is not directly applicable for general quadratic cost functions. Secondly, we are interested in the shape of the surface, not the number of correctly classified sign bits. A small number of misclassifications may have large influence on the shape or may not be visible at all (e.g. when p, q are both close to 0). Note that it is possible to have different solutions with very different shapes but similar objective values. It is also possible that the correct surface is not the minimal solution (e.g. when the correct surface is not smooth, or when there are insufficient or noisy constraints). This depends on the specific instance of the problem.

In our setting, due to the fact that the 0.878 approximation ratio does not apply directly, simple random hyperplane rounding typically requires a large number of iterations. We apply a series of heuristics for improvement and solution refinement:

1. Instead of picking plane normals from a uniform distribution on the sphere, we use the principle singular vectors of $\mathbf{R}$ (that correspond to largest singular values). For example, if we had to choose a single plane, a good choice for the normal would be the principle singular vector. Such "inertial" splitting methods have been previously used for other embeddings [22, 46], but to our best knowledge not for the SDP embedding. To widen the choices of planes, we randomly pick normals as a weighted linear combination of the singular vectors that correspond to the k-largest singular values, i.e.

$$\mathbf{N} = \sum s_i \lambda_i \mathbf{w}_i \ \ , \tag{4.11}$$

where $s_i \sim \mathcal{N}(0,1)$ and $\mathbf{w}_i$ is the singular vector corresponding to the singular value λ_i. Finding these singular vectors can be done efficiently by power-iteration methods.

2. Instead of making the decision based on a single normal, we randomly select a pair of normals $\mathbf{N}_1, \mathbf{N}_2$ according to (4.11), and perform a circular sweep for normals in the plane spanned by $\mathbf{N}_1, \mathbf{N}_2$. To do this, we project the points embedded in $\mathbb{R}^n$ on this plane (the first plane we check is the one spanned by the two principle singular vectors). Then we perform a circular sweep in this plane, as described in [18]. Basically, the sweep rotates a line through the origin that separates the points into two groups, and picks the partition with the lowest cost. We noticed that these angular sweeps can be made more efficient than described in [18]. Note that the cost of splitting n points in $\mathbb{R}^n$ based on a single random normal is $O(n^2)$. However, once the points have been projected on a plane and an initial cut has been computed (in $O(n^2)$), we can update the cost of the cut in $O(n)$ by careful bookkeeping similar to the Kernighan-Lin (K-L) algorithm [93]. As the line is rotated past a point, the only terms that need to be moved to the other side of the cut are the $n-1$ pairwise terms that involve that point (these terms are simply the corresponding row of $\mathbf{C}$ except the diagonal element). Thus, we can check n consecutive splittings also in $O(n^2)$, thereby reducing the amortized cost by a factor of n.

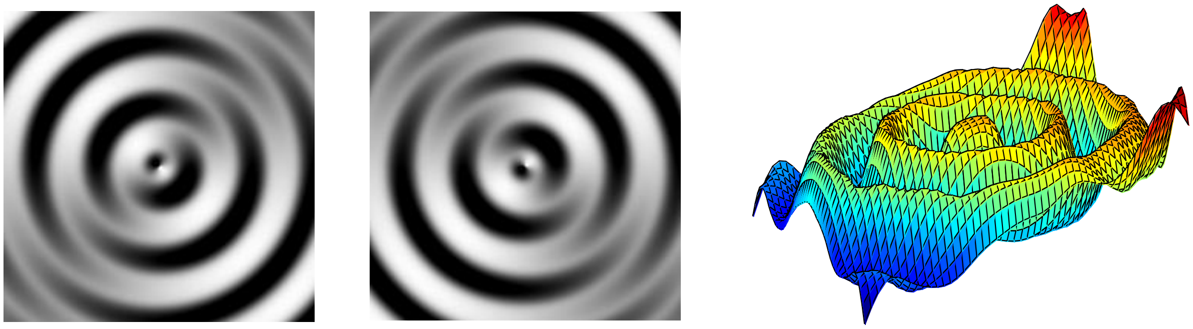


Figure 4.5: Surface computed from a synthetic photometric stereo pair.

3. The k-best results from the circular sweep phase are refined with the K-L algorithm [45, 88, 93, 110]. This is a local search procedure that will clean up a small number of misplaced vertices. We terminate this algorithm early if no progress is made in 50 consecutive iterations [88]. The lowest cost solution among these trials is returned.

4.4 Two-fold ambiguity results

Figure 4.3 demonstrates reconstruction from ambiguous normals. To simplify texture extraction we used square texture elements (see section 4.5 for derivation of normals from parallelograms). The SDP solver we used is DSDP [9].

In figure 4.5 we computed a surface from a pair of synthetic images of a Lambertian surface using two-light photometric stereo. The two possibilities for the surface normal are computed on a 29×29 grid. Points having only a single possible normal and points in attached shadow were removed. The system had 425 discrete variables. Our program made the correct decision at each of these points. However, the average deviation from the true surface is 19% (note that two corners are in shadow).

Results for segments of known length are shown in figures 4.4 and 4.6. In figure 4.4, the image was taken from a distance of about 10m with a zoom lens to approximate orthographic projection. As suggested by Naito and Rosenfeld [133], the 3D length of the segments is estimated as the maximum over the 2D lengths of all segments in the image. Figure 4.6 shows

1521 randomly oriented line segments tangent to a synthetic surface (top and bottom left). The top-center plot shows the projection of the SDP embedding on the subspace of the first two principle components. The splitting line is the lowest energy circular cut for this projection. To check the appropriateness of the projection on a low-dimensional subspace we projected the SDP embedding on the subspaces spanned by the first $1, 2, 3, \ldots$ principle vectors. The top-right plot is the squared-length of these projections, where the 1521 values are sorted. The lowest curve is the distribution of magnitudes of the projection on the subspace of the first principle vector; the second curve is the distribution of magnitudes for the projection on the subspace of the first two principle vectors, etc. It is evident that the points were embedded near a low dimensional subspace, and this is used for more efficient rounding. The solution (bottom-center) was computed using 300 spline bases, 1000 circular sweeps on random planes chosen by our algorithm, and K-L runs on the best 100 vectors. The output had 15 wrong flips, and the average height deviation from the truth surface is 1%. In comparison, a run of the Goemans-Williamson random hyperplane rounding (bottom right) with 10^4 trials produced 82 misclassifications, with 2.6% average height deviation (the reason we chose more random trials is that each iteration of the Goemans-Williamson rounding is faster than our rounding scheme). While these numbers depend very much on the particular instance, this example shows that the SDP approach to discrete ambiguities can deal with much larger instances than demonstrated by Forsyth [48].

4.5 Four-fold ambiguities

The previous sections looked at problems where each decision had two options. In this section we demonstrate an extension to discrete ambiguities with four options. As a model, we look at a shape from texture problem. Suppose a collection of similar triangles are scattered on a smooth surface viewed orthographically. All triangles are scaled versions of a known triangle. In addition, we assume that one edge on each triangle can be identified. For instance, if the

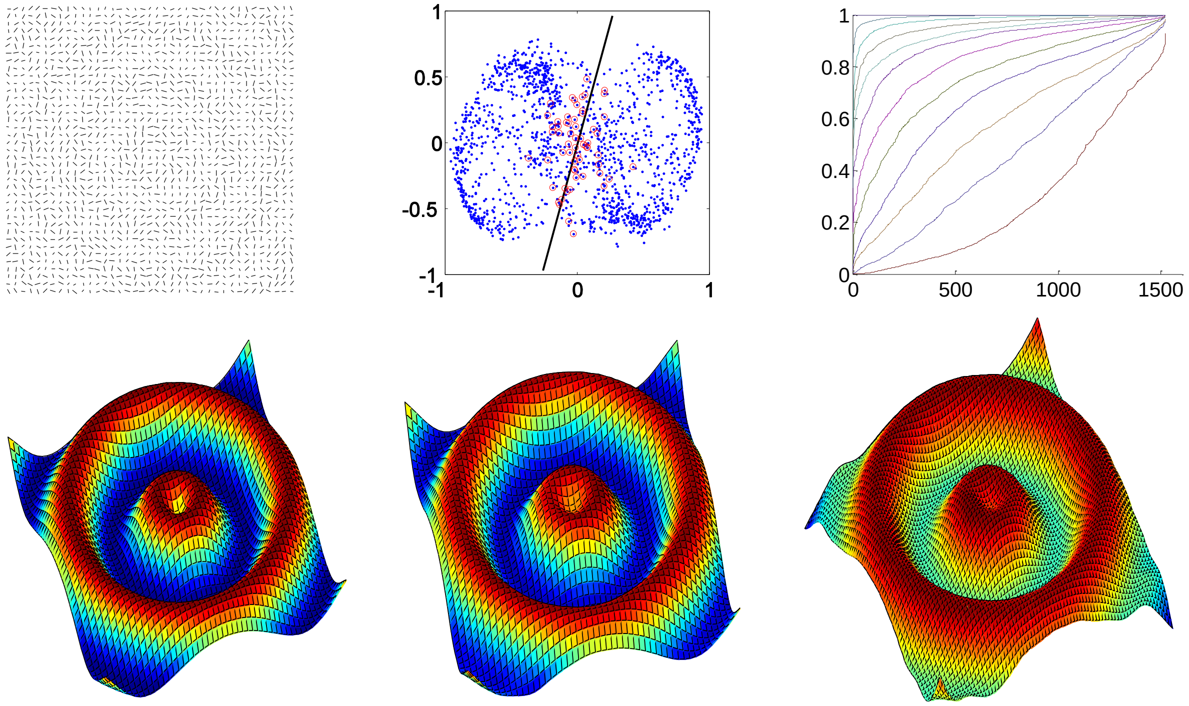


Figure 4.6: Top left: segments of known and equal length on synthetic surface. Top center: projection of the SDP embedding on the subspace of the first two principle components. Misclassified points with respect to the ground truth are circled. Top right: plot of the squared-length of the projections of the embedded points on the first principle subspaces (see text). Bottom left: original surface. Bottom middle: output of the program. Bottom right: output using random hyperplane rounding.

texture elements are rectangles, as in figure 4.7, we can identify the diagonal. This information leads to four-fold ambiguity since there are two ways to match the image segments with the edges of the known triangle.

Algebraically, let (x_i, y_i, z_i), $i = 1, 2, 3$, be the three vertices of a triangle in the image. Denote $dx_{ij} = x_i - x_j$, $dy_{ij} = y_i - y_j$, $dz_{ij} = z_i - z_j = p \cdot dx_{ij} + q \cdot dy_{ij}$, where p, q are the slopes of the triangle's plane. Since the triangle is similar to the model triangle, the 3D length ratios are known

$$r_1 = \frac{dx_{12}^2 + dy_{12}^2 + dz_{12}^2}{dx_{13}^2 + dy_{13}^2 + dz_{13}^2} \quad , \quad r_2 = \frac{dx_{23}^2 + dy_{23}^2 + dz_{23}^2}{dx_{13}^2 + dy_{13}^2 + dz_{13}^2} \quad . \tag{4.12}$$

This leads to two quadratic equations in p, q. Simple manipulations lead to a quadratic equation in q^2 that can be solved for the positive root (see Appendix B). Switching between r_1,r_2 gives another solution. In the general case, the four solutions are of the form $\pm(p_1, q_1)$ and $\pm(p_2, q_2)$.

In the SDP literature, the max-k-cut problem was studied by Frieze and Jerrum [52] and de Klerk, Pasechnik, and Warners [35]. While an ideal encoding requires two bits to encode four possibilities, their encoding uses four bits: a single indicator bit set to 1 and the rest 0. Since the matrix $\mathbf{X}$ has $O(n^2)$ entries, redundant encoding makes the SDP problem 4 times larger, which is a significant factor for current SDP solvers. There is a natural encoding of this problem with two sign bits for each constraint using average and offset vectors similar to the sums and difference vectors of section 4.2.2. However, we found that if the two bits are completely independent, the rounding phase becomes more difficult and results get worse. Instead, our encoding uses two variables d_1, d_2 for each triangle that ideally would take the values $-1, 0, 1$. We add the constraint $d_1 \cdot d_2 = 0$ so that only one decision variable is active at a time

$$p = d_1 p_1 + d_2 p_2 \quad , \quad q = d_1 q_1 + d_2 q_2 \quad , \quad d_1 \cdot d_2 = 0 \quad , \quad d_1^2 + d_2^2 = 1 \quad . \tag{4.13}$$

By expressing p, q for each triangle using the spline parameters and adding the smoothness term, we arrive at a system of the form (4.1) in $2n$ discrete variables. The SDP relaxation is modified to

$$\underset{\mathbf{X}}{\text{argmin}}\ \mathbf{C} \bullet \mathbf{X} \quad \text{s.t. } \mathbf{X}_{2i-1,2i-1} + \mathbf{X}_{2i,2i} = 1 \ \ , \ \ \mathbf{X}_{2i-1,2i} = 0 \ \ , \ \ \mathbf{X} \succeq 0 \ . \tag{4.14}$$

After solving the SDP problem (4.14) for $\mathbf{X}$, Cholesky factorization $\mathbf{X} = \mathbf{R}\mathbf{R}^t$ embeds the decision variables into a sphere in $\mathbb{R}^{2n}$. Let $\mathbf{u}_i$ denote the i-th row of $\mathbf{R}$, so that $\mathbf{X}_{ij} = \mathbf{u}_i \cdot \mathbf{u}_j^t$. For each decision there are two orthogonal vectors, $\mathbf{u}_{2i-1}$,$\mathbf{u}_{2i}$, such that $\|\mathbf{u}_{2i-1}\|^2 + \|\mathbf{u}_{2i}\|^2 = 1$. The rounding phase has to decide which of the two vectors is active, and round the active variable to either 1 or -1. In the ideal case, the vectors associated with the inactive variables would concentrate near the origin. While concentration can be observed, deciding which variable is active by picking the longer vector of each pair is not powerful enough. Instead, we execute the following heuristics:

1. A column of the matrix $\mathbf{X}$ is the inner product between a row vector $\mathbf{u}_i$ of $\mathbf{R}$ to the other rows. In the ideal case, the columns for inactive variables should be all zeros. We multiply $\mathbf{u}_i$ by α_i, the magnitude of the i-th column of $\mathbf{X}$. This has the effect of moving vectors orthogonal to the rest, or close to 0, towards the origin. In addition, the largest singular vectors of $\alpha_i\mathbf{u}_i$, that are used for projection on a plane for the circular sweep in (4.11), become less affected by the vectors of the inactive variables when these vectors are moved towards the origin.

2. The vectors $\alpha_i\mathbf{u}_i$ are projected on a plane as in (4.11), and a line is swept circularly. For each pair of indices $2i-1$ and $2i$, the point on the plane with the largest projection on the sweep line is considered active, and the sign of the projection is the rounded value (events in this circular sweep occur at angles where the projections of the points on the sweep line are equal in absolute value). We repeat this step for different planes and store the best circular cut for each plane.

3. For every variable, we estimate the probability p_i it is inactive as the percentage of best

cuts where it was inactive. Clearly, $p_{2i-1} + p_{2i} = 1$. We modify the diagonal of the matrix $\mathbf{C}$ to reflect this knowledge by setting $\mathbf{C}_{ii} = \mathbf{C}_{ii} + \mu p_i$ (μ is a tuning parameter). The SDP is solved a second time with the modified matrix $\mathbf{C}$. Due to the constraint $\mathbf{X}_{2i-1,2i-1} + \mathbf{X}_{2i,2i} = 1$, vectors which are believed to be inactive are pushed towards the origin.

4. We repeat steps 1,2 with the modified $\mathbf{C}$, keep the best solutions and run the K-L algorithm as a final step. The modification of the K-L algorithm to four possibilities is straightforward [214].

The method is demonstrated in figures 4.7 and 4.8. In figure 4.7, a collection of 415 similar rectangles at random orientations is overlaid on a 3D surface. Our program uses triangles made of the diagonal and two sides of each rectangle. Only the proportions of the model triangle are assumed known. The circular sweep in the plane of the first two principle components makes 148 errors in the first round (out of 830). After modifying the $\mathbf{C}$ matrix it makes 114 errors in the second round. Note that a large number of inactive variables are concentrated close to the origin and cannot be distinguished in this figure. The final computed surface (bottom middle) makes 80 errors. The average height difference between the original and computed surfaces is 5%. Wiggles in the computed surface arise because only 300 basis are used. Similar wiggles occur with a spline that uses the ground truth decision vector (bottom right). In this example, the cost function of the computed surface is lower than the cost of the ground truth spline.

4.6 Conclusions

Several depth cues, such as texture, shading and defocus, are inherently ambiguous at the local level. In this chapter we examined the integration of discrete constraints arising from these ambiguities with continuous objectives like surface smoothness. Problems of this form involve both continuous and discrete variables, and can be transformed into an entirely discrete optimization problem which is computationally hard. Following the approach of Zhu

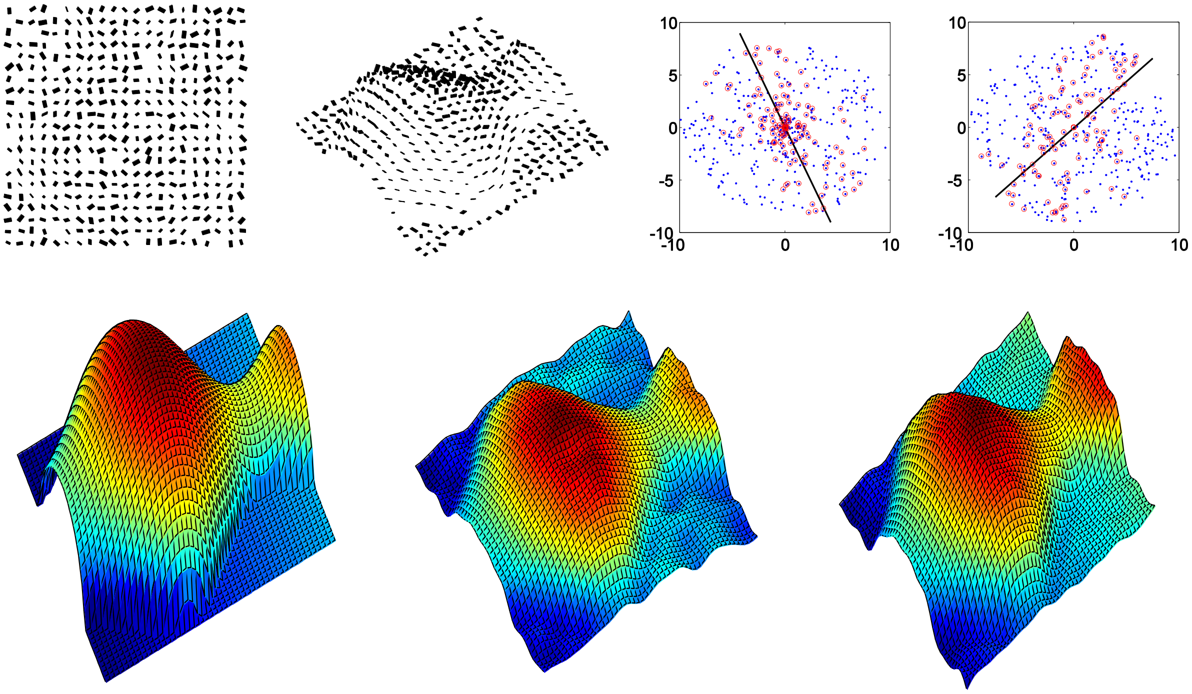


Figure 4.7: Reconstruction from similar rectangles. Top left: the input image is made of similar rectangles overlaid on a 3D surface. Top right: the circular cuts of the projections on the plane of the two principle directions for the first and second SDP embeddings. Misclassified points with respect to the ground truth are circled. Bottom left: original surface. Bottom middle: output of the program. Bottom right: spline surface using the ground truth decision vector.

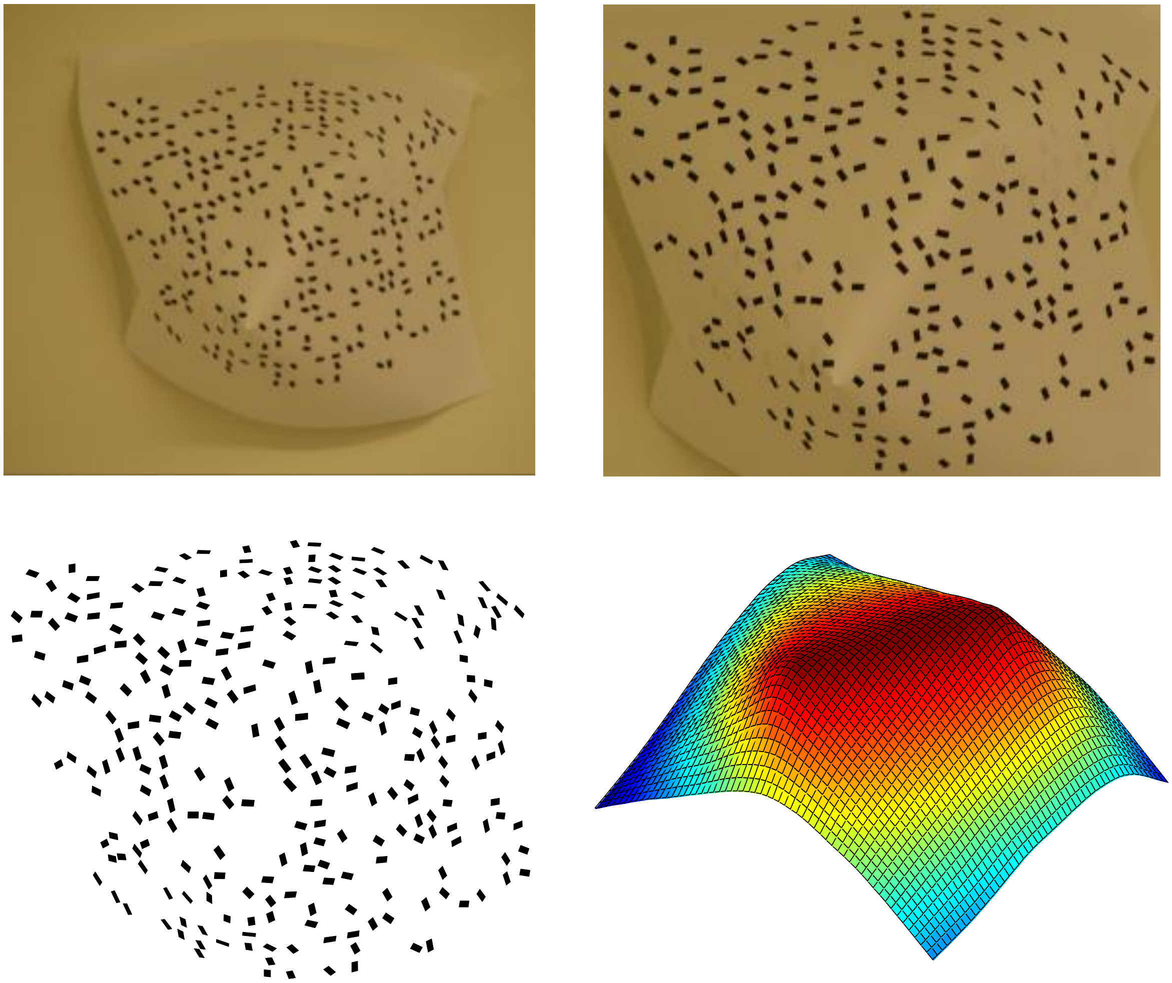


Figure 4.8: Top left: a surface textured with rectangles in ratio 1:2. Top right: the input image. Bottom left: parallelograms extracted from the input image. Bottom right: the computed surface.

and Shi [218], we presented a method to obtain an approximate solution using an SDP relaxation. We improved the rounding phase using a combination of heuristics, namely projection on planes in the subspace of the largest principle components, efficient circular sweep, and the K-L algorithm. These general heuristics were shown to be useful in our setting, and are potentially useful for other SDP applications as well. We conclude by briefly mentioning several possible extensions and alternative methods.

Our focus has been on developing a framework for solving problems of the form (4.1) that involve ambiguous discrete constraints. In practice, after binary decisions are made, the surface can be re-integrated using more robust integration methods. While the use of smoothness was demonstrated to resolve certain shapes, for many surfaces smoothness alone is insufficient and additional unambiguous constraints are required. A natural extension to the approach presented here would be to replace the spline with a shape basis, e.g. for particular shapes such as faces [3]. The general form of (4.1) allows for many other variations, such as adding linear constraints (e.g. specifying depths or normals at specific points [217]), or using shading information to disambiguate some normals [208].

Our formulation for shape from texture was under the assumption of orthographic projection. This assumption is reasonable, because texture elements are often seen through narrow angles. Two-fold ambiguities exist also in perspective projection. For example, suppose we observe an ellipse in the image which is the perspective projection of a slanted circle. If the camera is calibrated, it is possible to recover the normal to the circle's plane up to two-fold ambiguity [85]. Gårding [56] showed that to a first-order approximation, this two-fold ambiguity is a reflection of the normal around the line of sight. He also examined the global ambiguity of the surface that happens when all the normals are flipped around their line of sight. This ambiguity is analogous to the global reflection ambiguity in orthographic projection. Unlike the orthographic case, in perspective the two corresponding shapes will usually have different smoothness. The problem of integrating a smooth surface under perspective ambiguities seems to us more involved than the corresponding orthographic problem.

The main limitation of the proposed SDP scheme is its scalability. There are several ways to extend it to larger instances. First, there are large-scale SDP methods [131, 140]. Second, methods like CirCut [18] can be used to obtain high-quality cuts much faster than solving full SDP problems. The difference is that these methods depend on an initial guess. Third, it is possible to achieve scalability for the shape from texture problem, which is our main interest, using a hierarchical scheme, where the shape is solved up to in/out reversal over sub-windows. The solutions over these sub-windows can be glued together using the method of Zhu and Shi [218]. Fourth, it is possible to formulate the shape from texture problem using all the tilt angle constraints with coarsely sampled slant angle constraints. Recall from table 1.1 that tilt angles provide homogeneous linear constraints. These constraints do not differentiate between the two possibilities (p, q) and $(-p, -q)$ for the surface partial derivatives that can be recovered from the texture. Thus, a large number of tilt constraints could be added to the linear part of the system (our matrix $\mathbf{E}$), while the slant constraints could be sampled. Note that for a small number of slant constraints, e.g. 25, an exhaustive search over the flips is feasible, possibly with a final refinement step. However, for a moderate number of ambiguous constraints an exhaustive search is infeasible.

An important property of the SDP approach is that there is no need for initialization. However, if a good starting vector $\mathbf{d}$ is available, it can be exploited by performing the rounding phase on a linear combination of $\mathbf{X}$ and $\mathbf{d} \cdot \mathbf{d}^t$ [155].

Compared to discrete minimization approaches such as belief propagation, our approach is global, i.e. not based on local neighborhoods. Any discrete decision directly influences the cost of the entire surface. For a problem such finding a surface from segments of known length, knowledge of a label at a point could say little about a neighboring label. This property of the problem makes it different from other problems with discrete constraints. For instance, in segmentation problems there is often a strong correlation between nearby labels. A notable easy problem with discrete constraints is the interpretation of “Legoland” surfaces from cubic corners, which can be solved in linear time [97].

While we looked at discrete optimization, there is a complementary family of continuous optimization algorithms that could be applied. For the shape from texture problem, a continuous optimization formulation similar to (4.4) is

$$\underset{\mathbf{v}}{\operatorname{argmin}} \sum \left[((\mathbf{a}_{p_i} \cdot \mathbf{v})^2 - p_i^2)^2 + ((\mathbf{a}_{q_i} \cdot \mathbf{v})^2 - q_i^2)^2 \right] + \lambda \|\mathbf{E}\mathbf{v}\|^2 \ . \tag{4.15}$$

Forsyth [49] examined a similar expression with a different smoothness term. Similar expressions have also been encountered in graph drawing problems [55]. The problem (4.15) calls for continuous optimization in terms of $\mathbf{v}$. In chapter 6 we will discuss in more detail approaches for the minimization of quartic expressions, including iterative methods with exact line search and SDP relaxations. Broadly speaking, the discrete optimization approach has an advantage when the number of ambiguous discrete constraints is small. In the extreme case of a very small number of discrete constraints, an exhaustive search could be feasible, whereas iterative continuous optimization methods would not guarantee convergence to the global optimum. On the other hand, if the number of spline basis vectors is much smaller than the number of ambiguous discrete constraints, it might be more efficient to perform continuous optimization [205].

4.7 Appendix A: Our spline and smoothness term

In this section we describe in more detail the spline (4.2) and smoothness energy used in our implementation. These were chosen mainly for the sake of simplicity, and other splines and energies (e.g. thin-plate spline) could be used. To simplify notation assume the image is square. We used tensor-product spline bases

$$\mathbf{b}_{ij}(x, y) = \mathbf{b}_i(x)\mathbf{b}_j(y) \ , \tag{4.16}$$

where $\mathbf{b}_i, \mathbf{b}_j$ are singular vectors associated with small singular values of the matrix

$$\mathbf{D} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & & \cdots & 1 & -2 & 1 \end{bmatrix} . \tag{4.17}$$

For the smoothness energy we used the sum of squared second derivatives over the image. The energy of a basis function of the form (4.16), with $\|\mathbf{b}_i\| = \|\mathbf{b}_j\| = 1$, is

$$e_{ij}^2 = \|\mathbf{D}\mathbf{b}_i\|^2 + \|\mathbf{D}\mathbf{b}_j\|^2 \approx \iint \left(\frac{d^2\mathbf{b}_{ij}}{dx^2}\right)^2 + \left(\frac{d^2\mathbf{b}_{ij}}{dy^2}\right)^2 dx\, dy \ . \tag{4.18}$$

Note that for tensor-product splines we need to integrate only one-dimensional functions. The vectors $\mathbf{D}\mathbf{b}_i$ are proportional to the left singular vectors of $\mathbf{D}$ and hence orthogonal. Since the basis functions $\mathbf{b}_i$ are orthogonal, as are $\mathbf{D}\mathbf{b}_i$, the smoothness of a spline governed by $\mathbf{v}$ can be written as $\|\mathbf{E}\mathbf{v}\|^2$, where $\mathbf{E}$ is a diagonal matrix made of the elements e_{ij} for each basis used. The advantage of this approach over Fourier basis is that the basis vectors are not cyclic.

4.8 Appendix B: Normals computation for four-fold ambiguity

Consider a triangle with normal $(-p, -q, 1)$. Assuming the origin of the coordinate systems is at the centroid of the triangle, we can express its vertices as

$$P_1 = (x_1, y_1, px_1{+}qy_1) \quad , \quad P_2 = (x_2, y_2, px_2{+}qy_2) \quad , \quad P_3 = (x_3, y_3, px_3{+}qy_3) \quad . \tag{4.19}$$

Denote $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$, $L_{ij} = x_{ij}^2 + y_{ij}^2$, i.e. L_{ij} are squared lengths of the sides of the triangles in the image. The known squared ratios of the 3D lengths of the sides, r_1 and r_2, give the equations

$$\begin{cases} \frac{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}{(x_3-x_1)^2+(y_3-y_1)^2+(z_3-z_1)^2} = \frac{L_{12}+(px_{21}+qy_{21})^2}{L_{13}+(px_{31}+qy_{31})^2} = r_1 \\ \frac{(x_2-x_3)^2+(y_2-y_3)^2+(z_2-z_3)^2}{(x_3-x_1)^2+(y_3-y_1)^2+(z_3-z_1)^2} = \frac{L_{23}+(px_{23}+qy_{23})^2}{L_{13}+(px_{31}+qy_{31})^2} = r_2 \quad . \end{cases} \tag{4.20}$$

Arranging these equations as functions of p, q we can write

$$\begin{cases} A_1p^2 + B_1q^2 + C_1pq + D_1 = 0 \\ A_2p^2 + B_2q^2 + C_2pq + D_2 = 0 \end{cases} \tag{4.21}$$

Where

$$\begin{aligned} &A_1 = x_{21}^2 - r_1x_{31}^2,\ B_1 = y_{21}^2 - r_1y_{31}^2,\ C_1 = 2x_{21}y_{21} - 2r_1x_{31}y_{31},\ D_1 = L_{12} - r_1L_{13}, \\ &A_2 = x_{23}^2 - r_2x_{31}^2,\ B_2 = y_{23}^2 - r_2y_{31}^2,\ C_2 = 2x_{23}y_{23} - 2r_2x_{31}y_{31},\ D_2 = L_{23} - r_2L_{13}. \end{aligned} \tag{4.22}$$

To solve these two quadratic equations, first eliminate p^2 and express p in terms of q

$$\begin{aligned} &q^2(A_2B_1 - A_1B_2) + pq(A_2C_1 - A_1C_2) + (A_2D_1 - A_1D_2) = 0 \\ &p = \frac{q^2(A_2B_1-A_1B_2)+(A_2D_1-A_1D_2)}{q(A_1C_2-A_2C_1)}. \end{aligned} \tag{4.23}$$

Eliminate the pq term from the two equations

$$p^2(A_1C_2 - A_2C_1) + q^2(B_1C_2 - B_2C_1) + (D_1C_2 - D_2C_1) = 0. \tag{4.24}$$

Plug in p^2 as a function of q^2

$$\frac{(q^2(A_2B_1 - A_1B_2) + (A_2D_1 - A_1D_2))^2}{q^2(A_1C_2 - A_2C_1)} + q^2(B_1C_2 - B_2C_1) + (D_1C_2 - D_2C_1) = 0. \tag{4.25}$$

Let

$$\begin{aligned} &E = A_1C_2 - A_2C_1,\ F = B_1C_2 - B_2C_1,\ G = D_1C_2 - D_2C_1, \\ &H = A_2B_1 - A_1B_2,\ I = A_2D_1 - A_1D_2,\ t = q^2. \end{aligned} \tag{4.26}$$

We arrive at a quadratic equation in t, whose positive roots determine q and p

$$(tH + I)^2 + tE(tF + G) = t^2(H^2 + EF) + t(2HI + EG) + I^2 = 0. \tag{4.27}$$

Chapter 5

A Light Introduction to Shape from Shading

5.1 The standard shape from shading model

In this chapter we cover background material on shape from shading (SFS) as a preparation for chapter 6. Our goal here is not to provide a comprehensive literature review, but to highlight the main conceptual issues. General SFS surveys can be found in [75, 80, 147, 149].

While curves and texture provide constraints on 3D shape, it is important to understand that our picture of single-image 3D reconstruction would be incomplete without understanding SFS. Shading is important for depth perception in the interior of shapes far from their boundaries. When an image is reduced to a line drawing, the information on which side of an edge is dark and which side is bright is lost. Since the same image curves could arise from many different 3D shapes, shading information could reduce this ambiguity.

In computer vision, research on the SFS problem was pioneered by Horn [75], who formulated the problem mathematically as a differential equation and proposed the characteristic stripes method to solve it. The idea that shading is a cue for 3D perception is older. An historical account of related work prior to Horn is given in the last chapter of [75].

Horn's approach to SFS is to invert the physical process of image formation. The brightness of each image pixel is the result of an interaction (BRDF) between the light source direction and intensity, the surface orientation, the surface reflectance properties and the position and orientation of the camera. If the parameters of the image formation process are known, it might be possible to solve for the 3D shape of the surface.

The standard SFS model makes additional assumptions to simplify the problem. The surface is assumed to be textureless with constant albedo which is known. There are no visible regions of the surface which are in cast or attached shadow, and there is no inter-reflection of light between points on the surface. The model assumes a single, distant and uniform light source. The light source intensity and direction are assumed to be known. It assumes a distant viewer which translates to orthographic (weak perspective) projection. The reflectance of the surface is assumed to be Lambertian. This means that, due to the foreshortening effect, the intensity at a pixel is proportional to the cosine of the angle between the light source and the surface normal, independently of the viewing direction. Under these assumptions, the SFS problem is to estimate a surface that satisfies the SFS equation at each point

$$I = \rho \frac{\lambda \mathbf{L} \cdot \mathbf{N}}{\|\mathbf{N}\|} = \rho\lambda \frac{-ap - bq + c}{\sqrt{1 + p^2 + q^2}} \quad . \tag{5.1}$$

In equation (5.1), I is the image intensity at a point, ρ is the surface albedo, λ is the light source intensity, $\mathbf{L} = (a, b, c)$ is a unit vector in the light source direction, $\mathbf{N} = (-p, -q, 1)^T$ is the surface normal, where $p = \frac{dz}{dx}$ and $q = \frac{dz}{dy}$ are the partial derivatives of the surface $z(x, y)$.

It is sometimes assumed that the brightest point in the image corresponds to a normal pointing in the light source direction, and hence its intensity is $\rho\lambda$. When the image is scaled by $\rho\lambda$, for example by division by the intensity of the brightest point, the equation becomes

$$I = \frac{\mathbf{L} \cdot \mathbf{N}}{\|\mathbf{N}\|} = \frac{-ap - bq + c}{\sqrt{1 + p^2 + q^2}} \quad . \tag{5.2}$$

This is the form that we will be using.

In reality, whether or not the surface is Lambertian can be a function of scale. It could be that a blurred image of a textured surface, or an image taken from far away, will be approximately Lambertian, while a material that is considered Lambertian will look textured in a very high resolution image.

When there are several ideal light sources and all of them are visible in a region on a Lambertian surface, $\lambda\mathbf{L}$ is the linear sum of these light sources. Thus the assumption of a single light source is not too restrictive.

The special case where $\mathbf{L} = (0, 0, 1)$, i.e. the illumination comes from the viewing direction, leads to the Eikonal equation. The Eikonal equation can be written as

$$\frac{1}{I^2} - 1 = p^2 + q^2 \quad . \tag{5.3}$$

In this case there cannot be visible attached or cast shadows in the image (unless an occluding surface is between the camera and the light source). Note that the SFS equation with $\mathbf{L} = (0, 0, 1)$ is locally ambiguous, since there is an in/out reversal ambiguity for each local patch of any solution surface.

Several works studied existence and uniqueness of solutions to the SFS problem. Horn et al. and Brooks et al. [17, 76] demonstrated images for which there is no solution to the SFS equation. Oliensis [137] proved the uniqueness of SFS solutions when boundary conditions and other technical conditions are assumed. In absence of boundary conditions, the SFS problem is ambiguous. For example, consider the four surfaces $z = \pm x^2 \pm y^2$ with $\mathbf{L} = (0, 0, 1)$. In this case $p = \pm 2x$, $q = \pm 2y$, and therefore these four surfaces have exactly the same image. Saxberg [167] proved (under additional assumptions) that there are up to four local solutions near generic singular points (points whose normals are parallel to the direction of the light source). Okatani and Deguchi [136] showed that the second derivatives of the surface can be computed from image derivatives at singular points up to a discrete number of possibilities (subject to additional conditions). They argued that the singular points are the only points where it is possible to compute the second derivatives of the surface directly from local image derivatives. It should be noted that when the surface is not of class C^2 there might exist an infinite number of solutions. For instance, Deift and Sylvester [36] studied the ambiguities in the image of a Lambertian hemisphere. They showed that if the surface is C^2, the hemisphere is a unique solution up to a reversal ambiguity. Otherwise, there is an infinite number of solutions [108].

Also related to shading ambiguities is the GBR ambiguity, that we already encountered in chapter 2. When a surface undergoes a GBR transformation, its intensities will change, but for a small GBR transformation the change could be indistinguishable [7]. In summary, while there are theoretic conditions where the solution to the SFS problem is unique or there is only a discrete number of solutions, in practice the SFS problem is an ill-posed problem.

5.2 Classes of shape from shading methods

In this section, we review the main approaches for solving the SFS problem, without getting into the technical details. The survey paper by Zhang et al. [147] classifies SFS methods into four groups: minimization, propagation, local and linear. Most methods today are either minimization or propagation methods. These methods can be sub-classified according to the following criteria:

1. **Surface representation.** While so far we discussed surfaces of the form $z(x, y)$, or other linear surface representations, several SFS methods (e.g. [211]) solve for the normal parameters (p, q) with the integrability constraints $p_y(x, y) = q_x(x, y)$. This representation doubles the number of parameters and requires a final surface integration step. The possible benefit of this representation is that the algorithm can adjust the normals independently while relaxing the integrability constraints. It is not clear whether one representation is better than the other.

2. **Boundary and initial conditions.** Propagation and global minimization approaches

depend heavily on boundary and initial conditions. The two main classes of propagation approaches are those that begin from known boundary conditions [1, 87, 164] and those that begin from singular points [11, 96, 138, 200, 218]. Global minimization approaches are usually started from simple surfaces such as a plane, or from the intensity image [149]. The reason why initialization from intensities might work is that for some special surfaces, such as a sphere or a cylinder, the image of the surface when lit from the viewer direction is proportional to the depth of the surface.

3. **Regularization.** Regularization terms are often added to suppress high derivatives in the solution surface. Some propagation methods that start from known boundary conditions do not include regularization terms at all. Regularization might also be implicit in a scheme [1, 87]. In general, regularization makes a method more robust, but tends to over-smooth the surface.

Historically, Horn's characteristic stripes method was developed in full generality, before he simplified the problem to the orthographic Lambertian model. In recent years there has been renewed interest in lifting some of the assumptions of the simplified model. Methods for solving the SFS problem under perspective projection are described in [145, 150, 186, 216]. Note that some works on perspective SFS omit a $\cos^4 \alpha$ term, where α is the angle between a viewing ray and the image plane. This term is mentioned, for example, in [117], and in principle can be removed at the camera. The assumption that the light source is remote was replaced with a proximal light source in [149]. The problem of estimating the light source direction from an image was considered in several works, e.g. [75, 101, 177]. A very different approach for dealing with unknown illumination was taken by Smith [178], who derived a PDE with higher-order derivatives in which the light vector was eliminated. Smith was unable to solve his PDE, and his approach was not continued.

5.3 Human perception of shape from shading

Horn's influential SFS model is an elegant mathematical simplification. However, its assumption of Lambertian reflectance does not hold for many real-world objects. In contrast, it seems that humans do not have a special problem to perceive the 3D shape of many real-world non-Lambertian objects. In this section we review research on human shape perception from the psychophysics literature. Shape perception literature is too broad to be covered here in detail. Our focus here is on experiments that examine the assumptions of the standard SFS model. We caution the reader that human perception is not yet understood at the computational level. It is difficult to draw decisive conclusions from these experiments.

Note that humans cannot presume that the albedo is constant at every point. Research on how humans infer lightness (perceived reflectance or albedo) has a long history, that is covered in detail in Gilchrist's book [58]. Many factors seem to be involved in this non-local process, such as the intensity of the brightest point in the image, edge contrast, and even the perceived 3D structure of the scene. While we focus on images with constant albedo, it should be kept in mind that the brain executes another process, which is not fully understood yet, to interpret the raw brightness values of image pixels.

The perception literature considers shading as a weak depth cue that can be overridden by stronger cues. For instance, Ramachandran [153] gave an example where stereo overrides SFS perception, and another example where the same shaded image can be perceived in two different ways if the bounding contours are changed. These examples also demonstrate that SFS is a global process, since the same small image patches can be perceived differently in a different global context.

It is believed that SFS analysis is an early process in the brain. Kleffner and Ramachandran [98] showed that SFS can provide a basis for perceptual grouping of objects based on their perceived 3D structure. They demonstrated pop-out of a shape surrounded by differently shaded shapes. SFS can provide correspondences for motion analysis as well. Mamassian et al. [128] looked for areas in the brain that are active as SFS tasks are being processed. They found activity in early processing areas, and suggested that SFS is mostly a bottom-to-top process.

Several studies attempted to evaluate the accuracy of humans SFS perception. In a typical setting, participants were shown a shaded shape on a screen and asked to orient the normal vector at specific points on the surface. In a classical experiment by Koenderink et al. [105] it was found that the normals could be integrated to a real surface. Repeated experiments with the same subject on the same image correlated well. In a similar experiment [103], they noticed some differences between subjects and even between different trials of the same subject. They argue that these differences can be mostly attributed to scaling of the shape in depth. Another finding was that tilt estimate is more accurate than slant estimate. The slant was usually underestimated.

Other studies checked whether humans are assuming surfaces are Lambertian. Todd and Mingolla [75] experimented with ellipsoids rendered using Lambertian and non-Lambertian reflectance functions, highlights and cast-shadows. They found no difference in accuracy when shiny displays are compared to Lambertian displays. There was no difference when the ellipsoid had a cast shadow compared to the conditions when there was no shadow.

Langer and Bülthoff [112] studied human SFS perception under diffuse lighting (“cloudy day”) condition. In this condition, the amount of light a point receives is proportional to its visible fraction of the sky, and deep points are usually darker. Subjects were asked whether points on a surface are in a valley or on a hill under both diffuse and point light source conditions. Performance was well above chance, and the participants said they were unaware of the lighting condition changing between trials.

There is evidence that the light source direction is somehow encoded in the human visual system. Kleffner and Ramachandran [98] studied the perception of shaded half-hemispheres. In one experiment, they showed two rows of balls, where the rows were illuminated from opposite directions. One row was perceived as convex balls and the other as concave balls. It was impossible to perceive all the balls as convex simultaneously. When inversion occurred, it occurred globally. They concluded that there is a built in assumption of a single light source. They also argue that humans assume that light comes from above. When subjects were asked to tilt their heads by $90°$, a convex/concave inversion occurred. They reasoned that the "light-from-above" assumption is made in retinal (gaze) frame of coordinates without an adjustment of the head direction. Later, Perona and Sun [127,184] found a bias to assume that light comes from the top-left direction.

Boyaci and Maloney [15] demonstrated invariance of human SFS perception to monotone (not necessarily linear) transformation of the brightness values. To illustrate, consider the image in figure 5.1(A). In parts (B) and (C) the brightness values of this image were scaled to the range [0,1] and raised to the powers of 0.5 and 2 respectively. Notice that the perceived shapes outside the dark areas is very similar. The negative image in part (D) is not perceived as an ellipsoid. This is in agreement with the single light source assumption, since there is no "negative light" that will leave the shape intact, nor there is a single positive light source that will generate this image from an ellipsoidal shape. Non-monotonic brightness transformations alter the shape perception, as illustrated in parts (E) and (F).

5.4 Alternative shape from shading models

The presence of many non-Lambertian surfaces in the real world inspired further research on modeling the SFS problem. In this section we review some alternative research to the standard SFS model.

A natural candidate for study is the field of isophotes. Isophotes, or iso-intensity contours, are curves of constant brightness in the image. There are several reasons to study isophotes. Under the assumptions of the standard Lambertian model (section 5.1), the orientation θ of an isophote is independent of the albedo ρ and the illumination strength λ

$$\tan\theta = \frac{I_y}{I_x} = \frac{\rho\lambda\frac{d(\mathbf{L}\cdot\mathbf{N}/\|\mathbf{N}\|)}{dy}}{\rho\lambda\frac{d(\mathbf{L}\cdot\mathbf{N}/\|\mathbf{N}\|)}{dx}} = \frac{\frac{d(\mathbf{L}\cdot\mathbf{N}/\|\mathbf{N}\|)}{dy}}{\frac{d(\mathbf{L}\cdot\mathbf{N}/\|\mathbf{N}\|)}{dx}} \quad . \tag{5.4}$$

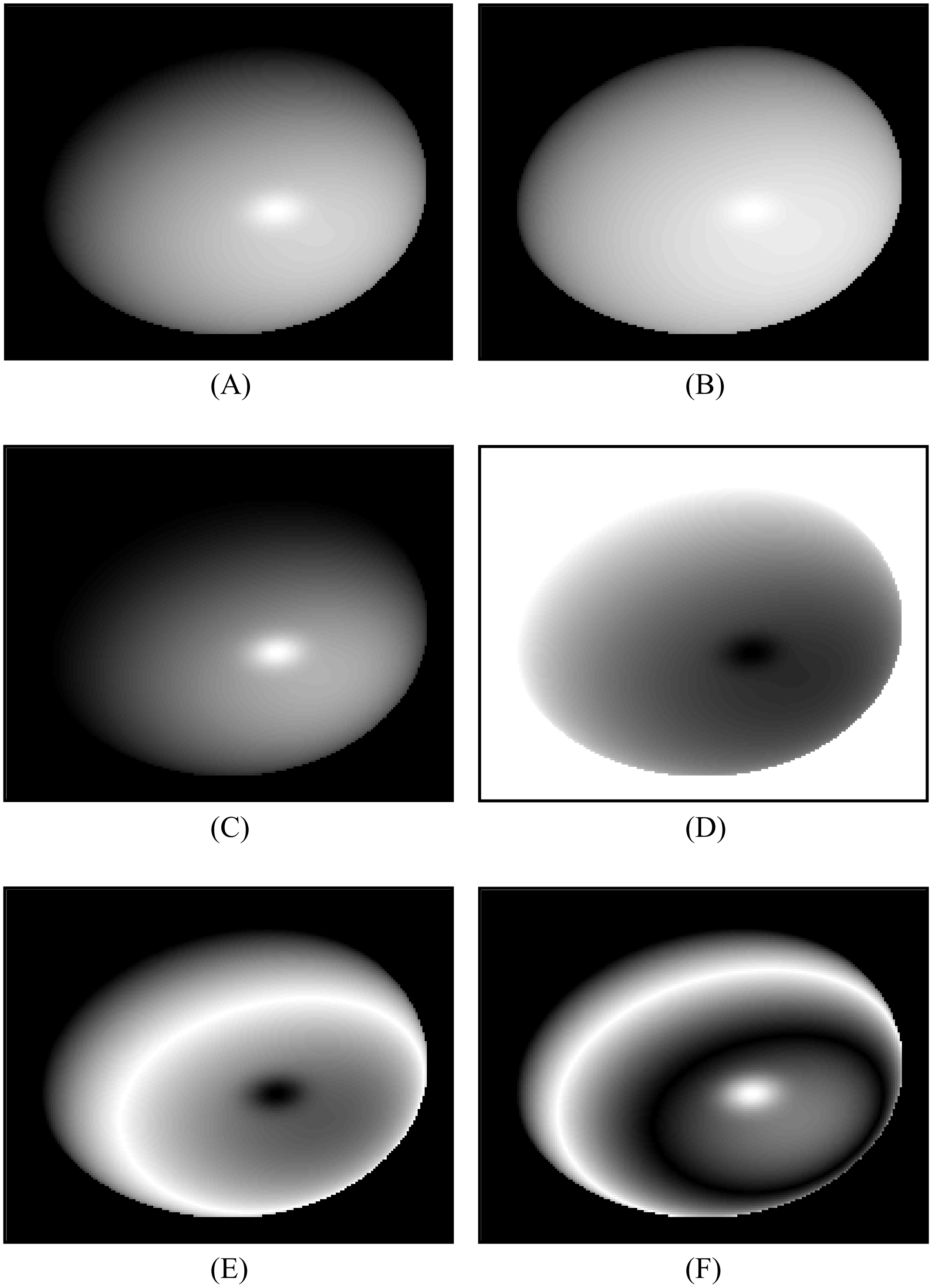


Figure 5.1: Effect of intensity transformations on perceived shape. When the image of a Lambertian ellipsoid (A) undergoes a monotonic transformation of its intensities, as shown in (B) and (C), the 3D shape looks similar. A negative transformation (D) and non-monotonic transformations (E) and (F) result in different perception.

One motivation to study isophotes is that the brain has orientation-selective cells [142]. Another reason is that monotonous brightness transformations, of the sort shown in figure 5.1, leave the field of isophotes intact. Isophote fields have been studied extensively and have many interesting properties [16,47,166,206,209]. None of these works was able to drop the assumption of known BRDF and propose an alternative way for complete 3D reconstruction directly from the field of isophotes.

Nevertheless, several shading models that have been proposed end up being a parametric transformation of the brightnesses of Lambertian images. These transformations are usually monotonic. When the parameters of the transformation are known, it is possible to invert the transformation and run a SFS algorithm on a Lambertian image. Some examples to such models are given below.

1. The reflectance of a Lambertian surface under hemispherical uniform illumination, assuming constant albedo and no inter-reflections or self-shadows, is given by [75]

$$I = \frac{L_0}{2}\left(1 + \frac{1}{\sqrt{1+p^2+q^2}}\right) \quad , \tag{5.5}$$

   where L_0 depends on the illumination strength and surface albedo. Note that the transformed image $2I/L_0 - 1$ will look like a standard Lambertian image illuminated from the viewer direction.

2. The direct reflection component from water under hemispherical uniform illumination is governed by the Fresnel coefficient [187]. A complete model for the reflectance of water should take into account the light coming from inside the water, shadows and inter-reflections which we neglect here. The Fresnel coefficient is a monotonic non-linear function of the angle between the observer and the surface normal. In contrast to the Lambertian case, water achieve minimal brightness when the surface normal points to the viewer, and maximal brightness at grazing angle. Thus, images of water are similar to negative images of Lambertian surfaces.

3. Images produced by a scanning electron microscope (SEM) are similar to reciprocal images of Lambertian surfaces. According to the theoretical model by Shatas et al. [170], the image intensity is proportional to $\frac{1}{\cos\theta}$, where θ is the angle between the surface normal to the microscope's primary beam. A similar empirical model is given by Ikeuchi and Horn [75].

5.5 Conclusions

The SFS problem involves modeling the image formation process, numerical optimization, and deciding among ambiguous solutions. In this chapter we took a closer look at Horn's Lambertian model. This model has always been considered as an idealized simplification, and its assumptions were sometimes regarded as unrealistic. However, all the alternative models that have been proposed so far might model particular scenes more accurately, but would be less accurate in other circumstances. A photorealistic modeling of the image formation process might be too complicated to invert. For instance, some models in computer graphics simulate subsurface scattering [81]. Another issue is that we have many possible models to choose from. Even worse, the scene might be composed of multiple objects that follow different reflectance models. As explained in the previous section, several models can be reduced to the Lambertian model after a suitable brightness transformation. Thus, it is possible that a brightness normalization process has to take place before running a SFS algorithm. In the next chapter we leave these modeling questions open, and move on to examine numerical optimization and ambiguities of the standard SFS problem.

Chapter 6

Polynomial Shape from Shading

6.1 Introduction

The shape from shading (SFS) problem [75, 147] is to recover the 3D shape of a surface from a single image, whose intensities are related to angles between surface normals and light source direction. SFS belongs to a wide class of problems in computer vision that involve embedding points in Euclidean space based on distance or angle information. These problems have natural formulations as systems of polynomial equations. Both exact methods, such as Gröbner basis and homotopy continuation, and convex relaxation techniques, have been applied to polynomial systems arising from such diverse problems as structure from motion [8, 20, 83, 132], camera calibration [43] and low-dimensional embedding [152, 204]. In this chapter we apply similar techniques to the SFS problem, which traditionally was treated mostly as a general nonlinear PDE that is notoriously difficult to optimize. While the polynomial formulation is not new (e.g. [145]), only recently theory and software for polynomial systems became widely available.

Throughout we will focus on the standard Lambertian model that was elaborated in section 5.1, i.e. orthographic projection, a known distant light source, unit albedo, but no other boundary conditions. Denote the unit light source vector by $\mathbf{L} = (a, b, c)$, and the surface normal by $\mathbf{N} = (-p, -q, 1)^T$. The Lambertian intensity at an unshadowed point is

$$I = \frac{\mathbf{L} \cdot \mathbf{N}}{\|\mathbf{N}\|} = \frac{-ap - bq + c}{\sqrt{1 + p^2 + q^2}} \in [0, 1]. \tag{6.1}$$

Squaring and rearranging we get the quadratic equation

$$(1 + p^2 + q^2)I^2 - (-ap - bq + c)^2 = 0. \tag{6.2}$$

To avoid solutions to $\mathbf{L} \cdot \mathbf{N}/\|\mathbf{N}\| = -I$ we add the constraint

$$-ap - bq + c \geq 0. \tag{6.3}$$

The SFS problem is to find a surface $z(x, y)$ with $p = \frac{dz}{dx}$ and $q = \frac{dz}{dy}$ that satisfies (6.2) and (6.3) for each point in the image.

The polynomial form (6.2) has several advantages over the quotient form (6.1). First, small polynomial systems can be completely solved. We demonstrate this in section 6.2, where we show that all solutions to the SFS problem at vertices of an ideal, generic polyhedron can be found by homotopy solvers for polynomial systems. Second, exact line searches are possible in the polynomial form, but require expensive bisections in other forms. In section 6.3 we demonstrate the effectiveness of exact line searches as part of an iterative method for SFS on a grid. Third, semidefinite programming (SDP) relaxations for polynomial systems can produce approximate solutions without requiring an initial guess. We derive an SDP relaxation for SFS in section 6.4. In the absence of boundary conditions, the SFS problem is known to be ambiguous [108]. In section 6.5 we propose a method for generating artificial shading ambiguities as illustrated in figure 6.1.

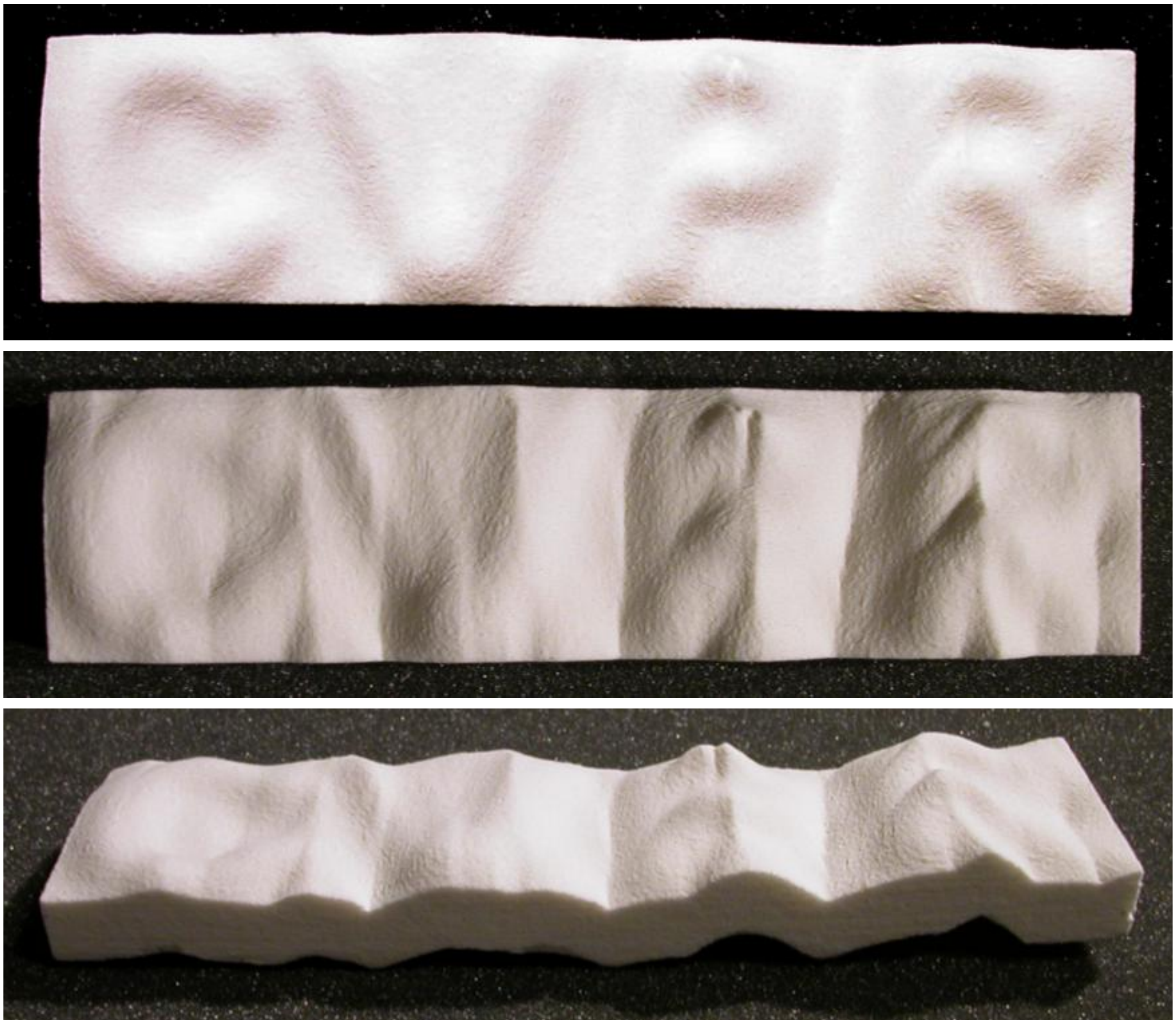


Figure 6.1: An exaggerated shading ambiguity illustrated by three images of one real object. The object has a pre-computed shape, designed using techniques described in this chapter, which was then realized by 3D-printing and illuminated with a directed light source (plus weak ambient). Bottom: an oblique view of the object. Middle and top: a top view, with the essential difference being that we moved the directed source to a different position. It is implausible that a human viewing the top image would infer the correct surface.

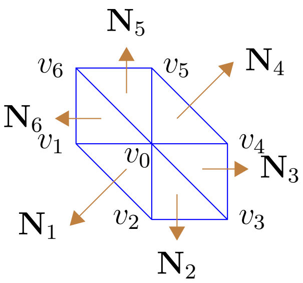


Figure 6.2: Triangulation around a vertex.

6.2 Exact SFS for polyhedra by homotopy

In this section we examine polyhedral SFS. The problem was first formulated by Horn [74]. Iterative procedures were proposed by Sugihara [183] and Lee and Kuo [116]. Penna [145] dealt with perspective projection. Yang et al. [213] highlighted the possibility of multiple solutions. Shimodaira [172] applied the DIRECT (dividing rectangles [82]) solver, which could be the only previous work with a solution guarantee. However, space-partitioning techniques tend to slow considerably as the dimension grows [179], require initial range estimation for the variables, and multiple solutions were not considered.

Since the problem can be formulated as a polynomial system, it is natural to apply homotopy solvers [179]. These solvers start with a structurally similar polynomial system for which all complex solutions are known, and trace the solutions as the system is continuously deformed to the desired polynomial system. Homotopy solvers guarantee "with probability 1" finding all complex solutions without requiring an initial guess, they scale with problem dimension better than space-partitioning methods, and are parallelizable. Modern solvers can be considered reliable, although rare breaking cases exist. See [179] for in-depth discussion and comparison to Gröbner basis and resultants. Previous applications of parameter continuation to SFS include real continuation [57], graduate decrease of smoothness [185] and continuation through scale space [156]. Watson [203] mentions early work related to homotopy and SFS that we were unable to find.

Assume the surface is a triangular mesh, and consider a particular vertex $\mathbf{v}_0$ and its k neighbors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ counter-clockwise as in figure 6.2. To simplify notation we shift the coordinates so that $\mathbf{v}_0 = (0, 0, 0)$. The normal to a triangle $(\mathbf{v}_0, \mathbf{v}_i, \mathbf{v}_{i+1})$ is proportional to $\mathbf{v}_i \times \mathbf{v}_{i+1} = (y_i z_{i+1} - y_{i+1} z_i, x_{i+1} z_i - x_i z_{i+1}, x_i y_{i+1} - x_{i+1} y_i)$. Dividing by the last coordinate we get

$$-p_i = \frac{y_i z_{i+1} - y_{i+1} z_i}{x_i y_{i+1} - x_{i+1} y_i} \quad , \quad -q_i = \frac{x_{i+1} z_i - x_i z_{i+1}}{x_i y_{i+1} - x_{i+1} y_i} \quad . \tag{6.4}$$

Substituting in (6.2) for every triangle, we get k quadratic equations in $z_1, \ldots, z_k$ (x_i,y_i are known from the image). If the system is generic, there are at most 2^k real solutions, which for small k can be found by a homotopy solver.

After solving for the possible configurations around each (internal) vertex, we prune the solutions. Configurations violating (6.3) or whose neighboring vertices have no compatible configuration are discarded. Then we need to solve a constraint satisfaction problem (CSP) to identify all consistent global solutions. Luckily, often an assignment of a configuration at a vertex and its neighborhood determines uniquely the configurations of its neighbors and therefore a simple assignment propagation and backtracking algorithm finds all global solutions.

Polyhedral SFS by homotopy is demonstrated in figure 6.3. The input contains 1143 triangles. Their vertices were randomly jittered to ensure generic systems. We used the homotopy solver HOM4PS-2.0 [118] to successfully find both solutions.

Note that theoretically it is also possible to solve for the light source, by writing the systems of the neighbors of a triangle together and forming a system of quartic equations with the condition $a^2 + b^2 + c^2 = 1$. This is practical only if the number of neighbors is small.

Unfortunately, the method described is sensitive to noise. Even 1% of intensities perturbation can lead to systems whose all solutions are complex. One might form the sum of squares of (6.2), take the partial derivatives and find all stationary points. This leads to k cubic equations, with up to 3^k solutions. Although solutions can be pruned, there still might be a large number of candidates. Given that for noisy systems it is harder to determine which neighboring configurations are compatible, the result is a much harder CSP.

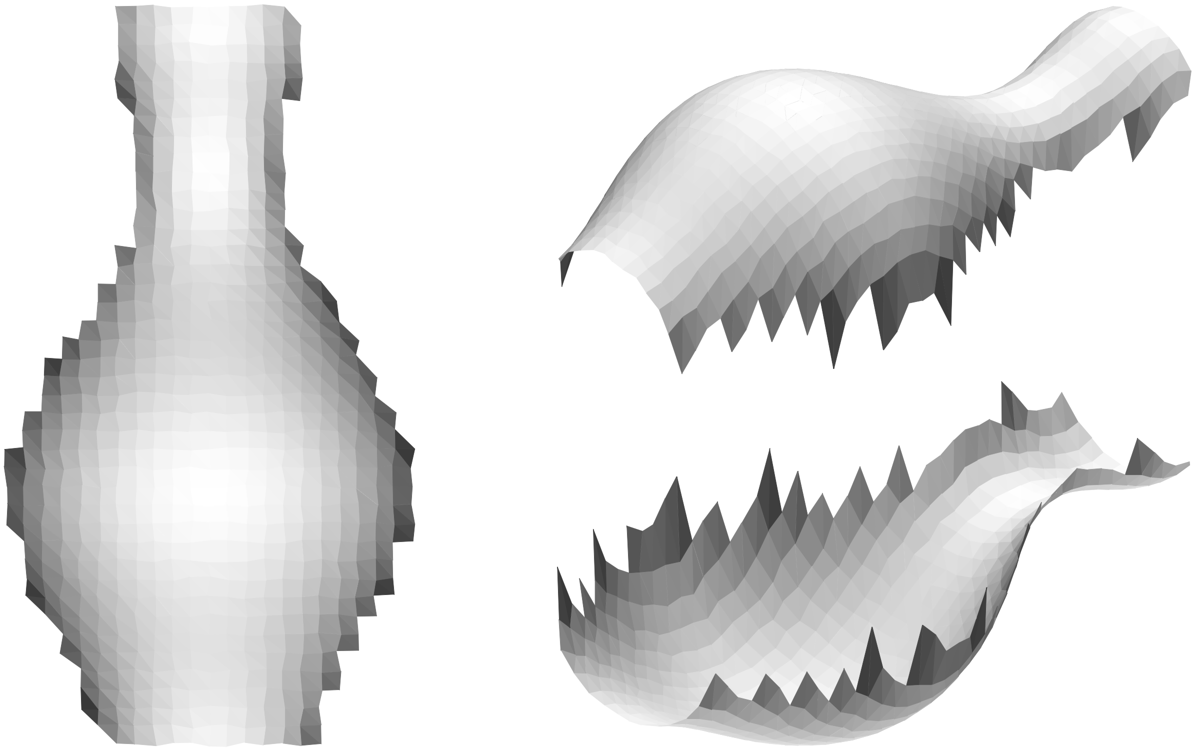


Figure 6.3: SFS of a polyhedron by homotopy. Left: synthetic Lambertian image with $\mathbf{L} = (0, 0, 1)$ of triangulated vase [147]. Right: the two solutions found by the algorithm.

The problem with noisy data should be no surprise. It is known that local SFS methods, such as Pentland's method [75] that assumes the surface is locally spherical, give unreliable normal estimates in presence of noise. The polyhedral case is similar, since image intensities information is available in a local neighborhood around each vertex. Furthermore, as the mesh resolution increases, we expect the intensities around a vertex to become nearly uniform, and hence the configuration around a vertex will inevitably become ambiguous. Note also that there is evidence that humans find SFS of triangulated meshes difficult [176]. However, in a controlled environment with a photometrically-calibrated camera, it might be possible to minimize the effect of noise by averaging the image intensities over each planar face.

6.3 Iterative procedure

In this section we present two technical improvements to global iterative SFS methods. These methods, e.g. [75, 147, 185], minimize the sum of squares of (6.2) over a grid. Denote by $\mathbf{z}$ the heights of all grid points arranged as a column vector, and let $\mathbf{p}_{ij} = \mathbf{z}_{i+1,j} - \mathbf{z}_{ij}$, $\mathbf{q}_{ij} = \mathbf{z}_{i,j+1} - \mathbf{z}_{ij}$ be the discrete partial derivatives of the surface. To avoid specifying any boundary conditions, for an $M \times N$ image we solve for $\mathbf{z}$ on an extended $(M+1) \times (N+1)$ grid (the $(M+1, N+1)$ pixel is redundant). Then (6.2) becomes

$$\begin{aligned}&\big[\mathbf{z}_{i,j}^2, \mathbf{z}_{i+1,j}^2, \mathbf{z}_{i,j+1}^2, \mathbf{z}_{i,j}\mathbf{z}_{i+1,j}, \mathbf{z}_{i,j}\mathbf{z}_{i,j+1},\\ &\mathbf{z}_{i+1,j}\mathbf{z}_{i,j+1}, \mathbf{z}_{i,j}, \mathbf{z}_{i+1,j}, \mathbf{z}_{i,j+1}, 1\big] \cdot \mathbf{u}_{ij} = 0 \quad ,\end{aligned} \tag{6.5}$$

$$\begin{aligned}\mathbf{u}_{ij} = \big[&2I_{ij}^2, I_{ij}^2 - a^2, I_{ij}^2 - b^2, 2a(a+b) - 2I_{ij}^2,\\ &2b(a+b) - 2I_{ij}^2, -2ab, -2c(a+b), 2ac, 2bc, I_{ij}^2 - c^2\big]^T \quad .\end{aligned} \tag{6.6}$$

Eq. (6.5) and (6.6) can be written in the form

$$\mathbf{r}_{ij} = \mathbf{z}^T \mathbf{A}_{ij} \mathbf{z} + \mathbf{e}_{ij}^T \mathbf{z} + h_{ij} = 0 \quad , \tag{6.7}$$

with symmetric $\mathbf{A}_{ij}$. Let $\mathbf{r}(\mathbf{z}) = \left[\mathbf{r}_{11}, \ldots, \mathbf{r}_{MN}\right]^T$. The sum of squared errors is

$$F(\mathbf{z}) = \|\mathbf{r}\|^2 \quad . \tag{6.8}$$

The Jacobian matrix of $\mathbf{r}$ has the rows

$$\mathbf{J}_{ij} = 2\mathbf{z}^T\mathbf{A}_{ij} + \mathbf{e}_{ij}^T \quad , \tag{6.9}$$

and the gradient is simply

$$\nabla F = 2\mathbf{J}^T\mathbf{r} \quad . \tag{6.10}$$

Minimization of a general multivariate quartic is NP-hard [124, 151]. However, a useful property of polynomials is that exact line search takes linear time. Substituting $\mathbf{z} = \mathbf{z}_0 + \alpha\mathbf{d}$ in $F(\mathbf{z})$, where $\mathbf{z}_0$ is the current point and $\mathbf{d}$ is any search direction, we get a univariate quartic in α, whose global minimization involves solving a cubic equation in α. Experience in numerical analysis suggests that exact line search often requires less function evaluations and a smaller number of iterations than inexact line search procedures. In addition, the algorithm is less likely to be trapped in a poor local minimum. Although very natural and easy to implement, we are not aware of previous applications of exact line search to SFS, or embedding problems like [204].

A smoothness term is often added to widen the basin of attraction for convergence and suppress oscillations. The most common regularizer measures thin-plate energy, which integrates the squared second derivatives of the surface. A problem with this regularizer is that it will strive to flatten regions of varying intensities or edges where the surface cannot be flat. In [63,77] it is proposed to down-weight the smoothness term by a measure of intensity change. That smoothness term frees the surface to bend where intensity is varying, but does not enforce folding explicitly. In addition, reducing the smoothness term allows surface oscillations, as will be explained in section 6.5.

Instead we define a new smoothness term that mitigates these problems. Consider two neighboring pixels with normals $(-p_1,-q_1,1)$, $(-p_2,-q_2,1)$ and intensities I_1,I_2. The smallest possible angle θ between these normals satisfies

$$\begin{aligned}\cos(\theta) &= \cos(\arccos(I_1)-\arccos(I_2)) \\ &= I_1\cdot I_2+\sqrt{1-I_1^2}\cdot\sqrt{1-I_2^2} \quad .\end{aligned} \tag{6.11}$$

For nearby pixels on a smooth surface, and for pixels with large intensity difference, it is reasonable to expect that the angle between the normals is close to the minimal possible θ, i.e.

$$\cos(\theta)\approx\frac{p_1p_2+q_1q_2+1}{\sqrt{1+p_1^2+q_1^2}\sqrt{1+p_2^2+q_2^2}} \quad . \tag{6.12}$$

Plugging in the Lambertian assumption (6.1) and multiplying by the denominator we define the smoothness term as another quartic in $\mathbf{z}$

$$S(\mathbf{z})=\sum\Big((p_1p_2+q_1q_2+1)I_1I_2-\cos(\theta)(-ap_1-bq_1+c)(-ap_2-bq_2+c)\Big)^2 \quad . \tag{6.13}$$

Our complete algorithm minimizes $F(\mathbf{z})+\lambda S(\mathbf{z})$ using conjugate gradient with exact line search, and reduces λ gradually. The method is demonstrated in figures 6.4 and 6.5.

In the algorithm above we ignored the non-negativity constraint (6.3). It is possible to take it into account by minimizing

$$\sum\Big((1+p^2+q^2)I^2-(-ap-bq+c)^2\,\mathrm{sign}(-ap-bq+c)\Big)^2 \quad . \tag{6.14}$$

The exact line search requires sorting α according to where the terms switch signs, and searching for the minimum of the quartics in each intermediate interval. The coefficients of these quartics can be accumulated as the line is traversed. This usually makes insignificant difference for images without shadows.

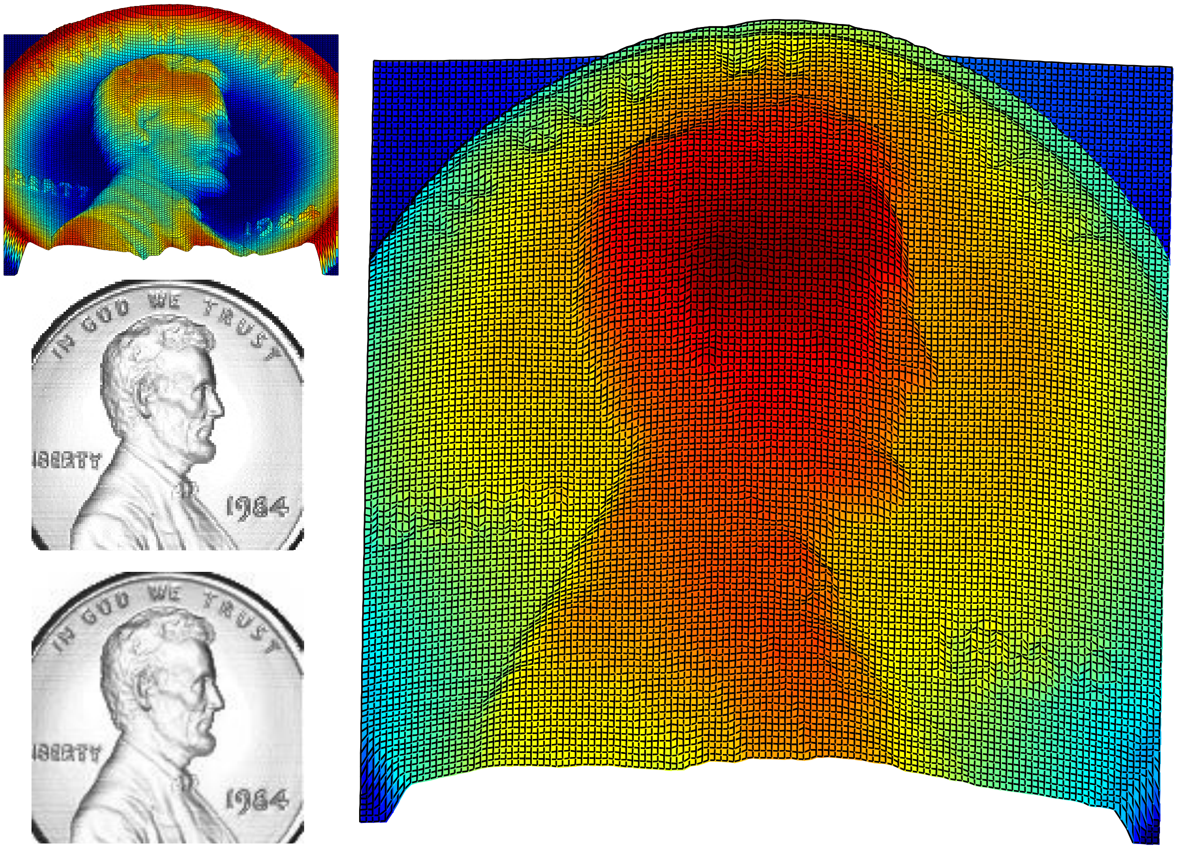


Figure 6.4: SFS of the synthetic penny [147]. The source surface (upper-left), of size 128×128, generates the input image (middle-left) with $\mathbf{L} = (0, 0, 1)$. Our iterative method, initialized from a section of a sphere, produced the surface on the right. The computed surface generates the lower-left image. While the 3D surfaces are different, the RMS of the images difference is only 0.008, and the maximal absolute difference of intensities is 0.117.

6.4 SDP relaxation

While iterative techniques with exact line search are effective, often there exist initial guesses that lead to suboptimal local minima. This motivated us to look at the completely different approach of convex relaxations, which guarantee convergence to a global minimum of a related problem. Convex relaxations have been recently applied to embedding problems from geometric measurements of distances and angles. For instance, Biswas et al. [12, 13] applied SDP relaxations to localization problems in sensor networks. We can think of the SFS problem as a similar geometric embedding problem, where the goal is to embed the points in depth such that the angles between the surface normals and the light source direction comply with the image brightness values.

In this section we derive a Lasserre-type SDP relaxation to SFS. Comprehensive theoretical background on the topic can be found in [114]. The basic idea is to introduce new variables for products of variables (lifting) so that non-convex terms become linear. Additional positive-semidefinite constraints are added to capture relationships between all variables and tighten the approximation. Since full relaxations are very expensive, scalability is achieved via sparse relaxation [125, 201]. The semidefinite constraints are defined on cliques, which in our case are just unit triangles.

We begin by defining the notation for the independent variables of the problem. For each point (i, j) we use an extended set of variables: $\mathbf{x}_{ij}^{rst}$ ideally corresponds to a monomial $(\mathbf{z}_{ij})^r \cdot (\mathbf{z}_{i+1,j})^s \cdot (\mathbf{z}_{i,j+1})^t$. Here, r,s,t are nonnegative integers, $r + s + t \leq 2d$, and d is called the relaxation order. In this notation $\mathbf{x}_{ij}^{k00}$,$\mathbf{x}_{i-1,j}^{0k0}$ and $\mathbf{x}_{i,j-1}^{00k}$ refer to the same variable corresponding to $(\mathbf{z}_{i,j})^k$, $k = 1$ refers to the surface height at point (i, j), and $\mathbf{x}_{ij}^{000}$ is the constant 1. These variables are arranged into moment matrices as follows.

Let $(r_1, s_1, t_1) = (0, 0, 0)$, $(r_2, s_2, t_2), \ldots, (r_D, s_D, t_D)$ be an enumeration of all r, s, t such that $r + s + t \leq d$ and $D = \binom{d+3}{3} = \mathrm{O}(d^3)$. For each image pixel, we would ideally like to represent the constraint that the following matrix

$$\left[1, \mathbf{x}_{ij}^{r_2 s_2 t_2}, \ldots, \mathbf{x}_{ij}^{r_D s_D t_D}\right]^T \cdot \left[1, \mathbf{x}_{ij}^{r_2 s_2 t_2}, \ldots, \mathbf{x}_{ij}^{r_D s_D t_D}\right] \tag{6.15}$$

is a positive semidefinite rank-one matrix. This condition is not convex in the problem variables. Instead, we use a convex relaxation. For each image pixel, a positive semidefinite constraint is defined on a $D \times D$ moment matrix whose elements are independent variables

$$\left[\mathbf{M}_{ij}\right]_{m,n} = \mathbf{x}_{ij}^{r_m+r_n, s_m+s_n, t_m+t_n}, \quad \left[\mathbf{M}_{ij}\right]_{1,1} = 1 \quad . \tag{6.16}$$

Ideally, $\mathbf{M}_{ij}$ would be equal to (6.15).

The SDP relaxation for non-shadowed pixels is

$$\min\left(\sum_{ij} \operatorname{trace}(\mathbf{M}_{ij}) + G \cdot \varepsilon\right) \quad \text{s.t.} \tag{6.17}$$

$$(a+b)\mathbf{x}_{ij}^{100} - a\mathbf{x}_{ij}^{010} - b\mathbf{x}_{i,j}^{001} + c \geq 0 \tag{6.18}$$

$$\begin{aligned} -\varepsilon \leq \big[&\mathbf{x}_{ij}^{r+2,s,t}, \mathbf{x}_{ij}^{r,s+2,t}, \mathbf{x}_{ij}^{r,s,t+2}, \\ &\mathbf{x}_{ij}^{r+1,s+1,t}, \mathbf{x}_{ij}^{r+1,s,t+1}, \mathbf{x}_{ij}^{r,s+1,t+1}, \\ &\mathbf{x}_{ij}^{r+1,s,t}, \mathbf{x}_{ij}^{r,s+1,t}, \mathbf{x}_{ij}^{r,s,t+1}, \mathbf{x}_{ij}^{r,s,t}\big] \cdot \mathbf{u}_{ij} \leq \varepsilon \end{aligned} \tag{6.19}$$

$$\sum_{ij} \mathbf{x}_{ij}^{100} = 0 \tag{6.20}$$

$$\mathbf{M}_{ij} \succeq 0, \quad \left[\mathbf{M}_{ij}\right]_{1,1} = 1. \tag{6.21}$$

In the system above, equation (6.18) is the constraint (6.3) written using discrete differences for p, q. Equation (6.19) is derived by multiplying (6.5) with monomials $(\mathbf{z}_{ij})^r \cdot (\mathbf{z}_{i+1,j})^s \cdot (\mathbf{z}_{i,j+1})^t$ for all nonnegative integers r, s, t such that $r + s + t \leq 2d - 2$. $\mathbf{u}_{ij}$ are vectors of constants as in (6.6), and ε is a slack variable. Condition (6.20) fixes the global depth ambiguity in orthographic projection. Equation (6.21) is the positive semidefinite constraint from (6.15).

The objective (6.17) minimizes the sum of traces and G times the slack variable ε. G is the total number of diagonal elements in all moment matrices, which is a constant for a specific problem. Ideally, the matrices $\mathbf{M}_{ij}$ should be rank-one. Trace minimization is a commonly used approximation to rank minimization [44].

The resulting SDP problem is solved in polynomial time without an initial guess by a SDP solver. Currently we extract the solution simply from $\mathbf{x}_{ij}^{100}$. Note that this extraction procedure could fail in case there are multiple solution. Since the problem is convex, the matrices $\mathbf{M}_{ij}$ could be convex combinations of solutions. For example, when $\mathbf{L} = (0, 0, 1)$, a combination between the solutions $z(x, y)$ and $-z(x, y)$ yields $\mathbf{x}_{ij}^{100} = 0$. A similar solution extraction with a random perturbation is described in [201], and applied to PDEs with boundary conditions [125]. Also note that Lasserre's condition for convergence as $d \to \infty$ in [113] does not hold for our sparse relaxation. However, only small orders d are practical anyways. A solution extraction scheme for dense relaxations is described in [69], but does not extend easily to sparse relaxations with noisy input.

The iterative and SDP procedures are compared in figure 6.5 on real images of cloth (upper two) and sugar (lower three). The intensities were linearly transformed to the range [0,1]. Since the light source direction is unknown, we sampled 100 light directions using the spherical spiral method [6], ran the iterative algorithm, and kept the best light for each image. Though more expensive, this approach is more robust compared to estimating the light directly from the images or adding the light as a parameter to the optimization. The initial state for the iterative method was $\mathbf{z} = 0$. The SDP results were obtained from low-resolution images (18×24) using the same light sources and relaxation order $d = 2$. The SDP solver we used is DSDP5.8 [9], interfaced with YALMIP [123]. For the iterative method, the RMS errors in reconstructing the input images are all below 0.01. RMS errors for the SDP method are (top to bottom): 0.06, 0.07, 0.09, 0.07 and 0.14.

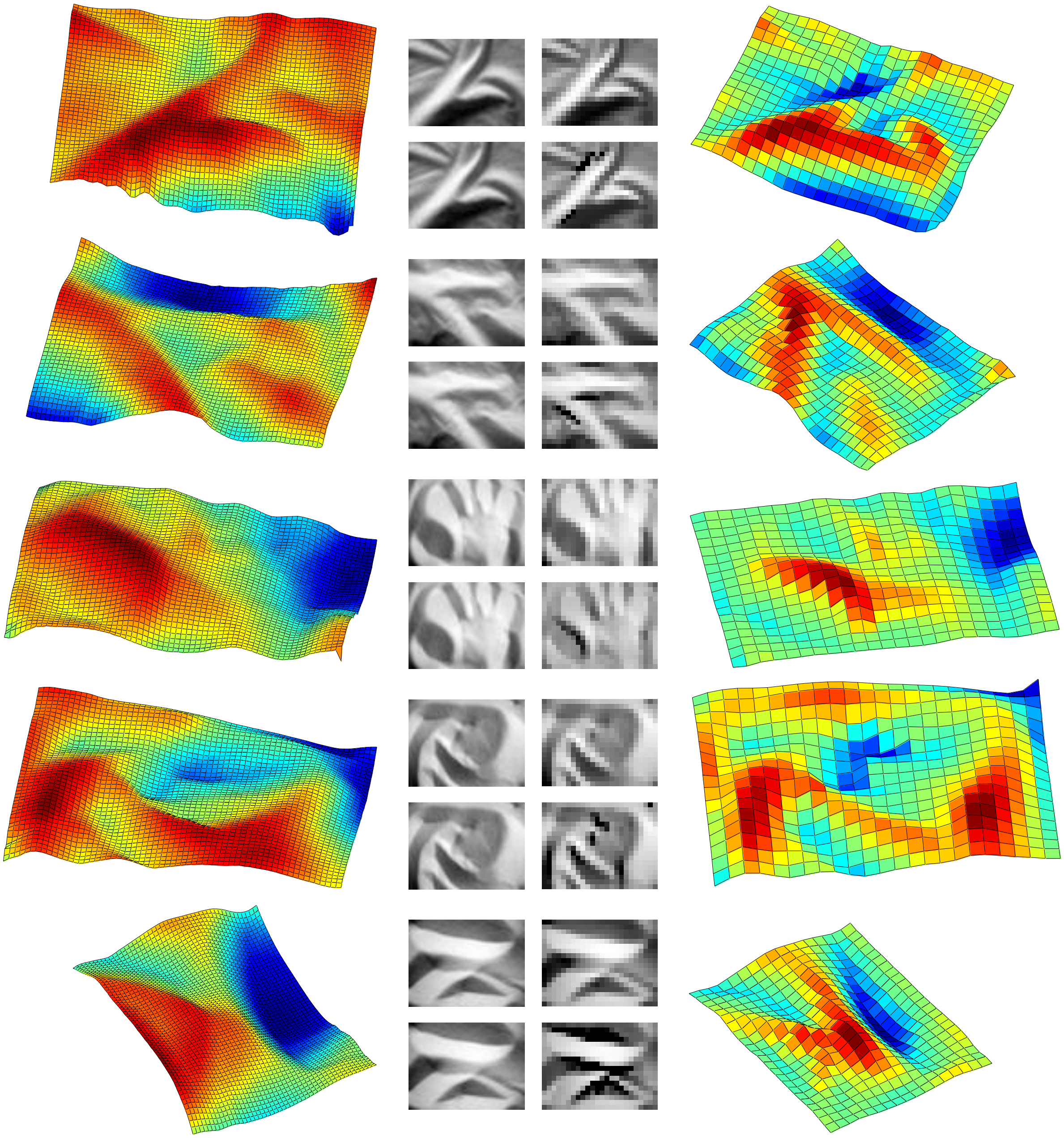


Figure 6.5: Surface reconstruction by the iterative (left) and SDP (right) methods. In the middle, the top two images are source images. The bottom images are renderings of the left and right surfaces. At black points in the SDP solutions condition (6.18) equals zero.

6.5 Generating SFS ambiguities

In this section we explore generating SFS ambiguities, i.e. substantially different surfaces that have identical Lambertian images. Our main goal is to provide researches a better way to study surface priors. Instead of defining a shape prior and running a SFS algorithm, a good prior should first be able to pick the correct surface from its ambiguous counterparts. We can also envision a future application in surface inspection, where by inspecting the space of ambiguous shapes, possibly with few known anchor points, one can position the light source so that ambiguous deviations are tolerable.

Previously, SFS ambiguities were encountered when SFS algorithms converged to the wrong surface, or in mathematical analysis of simple shapes such as planes and spheres [38, 108]. A recent work by Kemelmacher-Shlizerman et al. [92] studied Mooney faces generated by thresholding Lambertian images. They showed that different surfaces can have an identical isophote (iso-intensity contour) and an identical binary image. Another related ambiguity is the bas-relief [7, 207], which technically involves small changes in albedo.

SFS ambiguities can arise from micro-perturbations, e.g. placing a pyramid reflecting the desired intensity at every pixel [5]. These surfaces are uncommon. On the other extreme, smooth surfaces can also be ambiguous. For instance, a single-intensity image can arise from a plane or a cone whose axis is aligned with the light direction. Moreover, such cones can be reflected and stitched together, as shown in figure 6.6, creating a wavy surface. In general, a uniform-intensity image comes from a ruled surface [108]. Note that these surfaces can be very smooth. Furthermore, Freeman's generic light source assumption [50] cannot distinguish between a plane and such a cone.

For any surface whose image is clipped to just a patch, the SFS equations represent a PDE without boundary conditions. The lack of these boundary conditions gives rise to ambiguities. These ambiguities have been examined in the past for the continuous SFS equation [108]. We show below these ambiguities exist in the discrete case using the implicit function theorem.

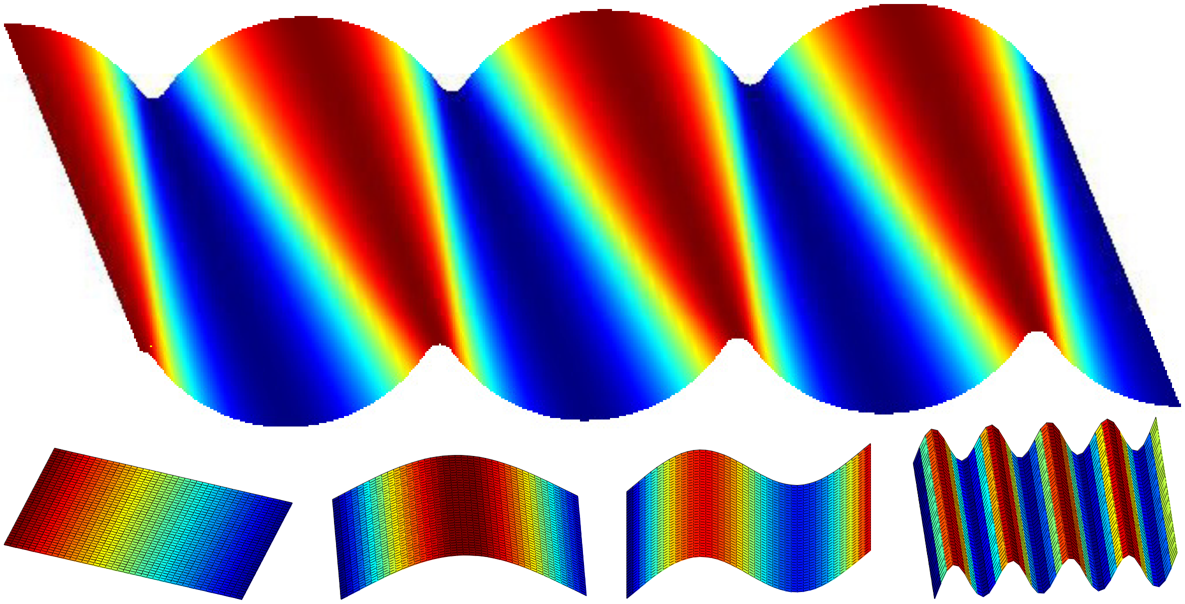


Figure 6.6: Shading ambiguities of uniform-intensity images. Top: interlaced cone sections with the same opening angle whose axes point in the light direction will generate a uniform intensity image. Bottom: some eigenvectors of $\mathbf{J}$ for the flat surface $\mathbf{z} = 0$ with $L = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$.

Theorem 1 (Implicit function theorem) *Suppose that* $G : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ *is continuous and has continuous partial derivatives. Suppose that* $G(x_0, y_0) = 0$ *and* $\det\left[\frac{dG^i}{dy_j}(x_0, y_0)\right]_{n\times n} \neq 0$. *Then there exists a neighborhood* $U \subset \mathbb{R}^m$ *of* x_0 *and a function* $g : U \to \mathbb{R}^n$ *such that* $G(x, g(x)) = 0 \quad \forall x \in U$. *Furthermore,* g *and its first partial derivatives are continuous on* U.

Consider a solution $\mathbf{z}_0$, i.e. $F(\mathbf{z}_0) = 0$ and $\mathbf{r}(\mathbf{z}_0) = \left[0, \ldots, 0\right]^T$. We will apply the implicit function theorem to the function $\mathbf{r}(\mathbf{z}) : \mathbb{R}^{(M+1)(N+1)} \to \mathbb{R}^{MN}$. Note that the Jacobian matrix $\mathbf{J} = \frac{d\mathbf{r}}{d\mathbf{z}}(\mathbf{z}_0)$ in (6.9) is $MN \times (M+1)(N+1)$. Assume further that $\mathbf{J}$ has full column rank. Then we can choose MN independent columns of $\mathbf{J}$ to form the "y" part in the implicit function theorem, and the rest of the columns will form the "x" part. The implicit function theorem implies that there exists an $(M + N + 1)$-dimensional manifold of solutions in an ε-neighborhood of $\mathbf{z}_0$. While the theorem guarantees solutions only in arbitrarily small neighborhoods, in many cases the ambiguity can be very substantial (see figures 6.1,6.7 and 6.8).

Next we describe how to generate numerically an ambiguous surface $\mathbf{z}$ from a given surface

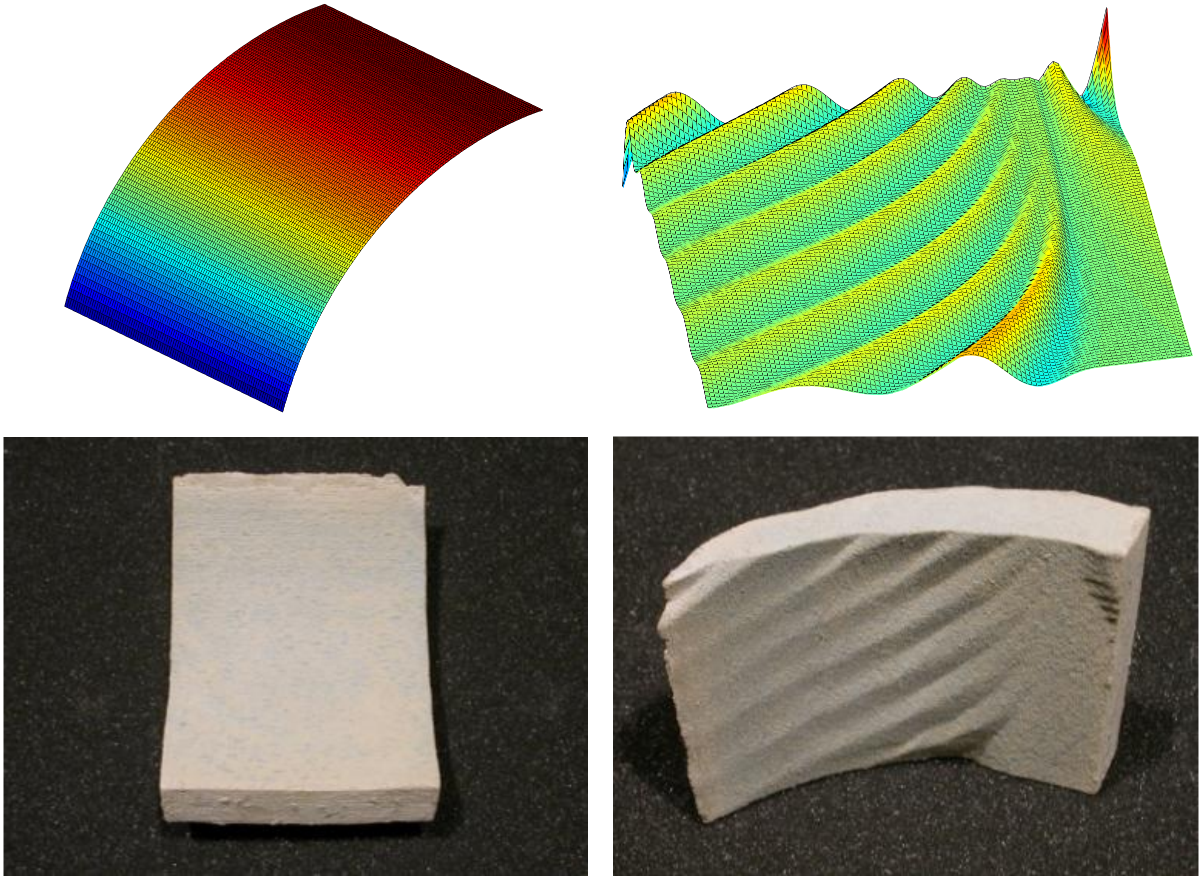


Figure 6.7: Artificially generated shading ambiguity. Top left: initial cylindrical surface. Top right: the null vector of $\mathbf{J}$. Bottom: 3D-print of the computed surface (right) generating the same image of a cylinder (left).

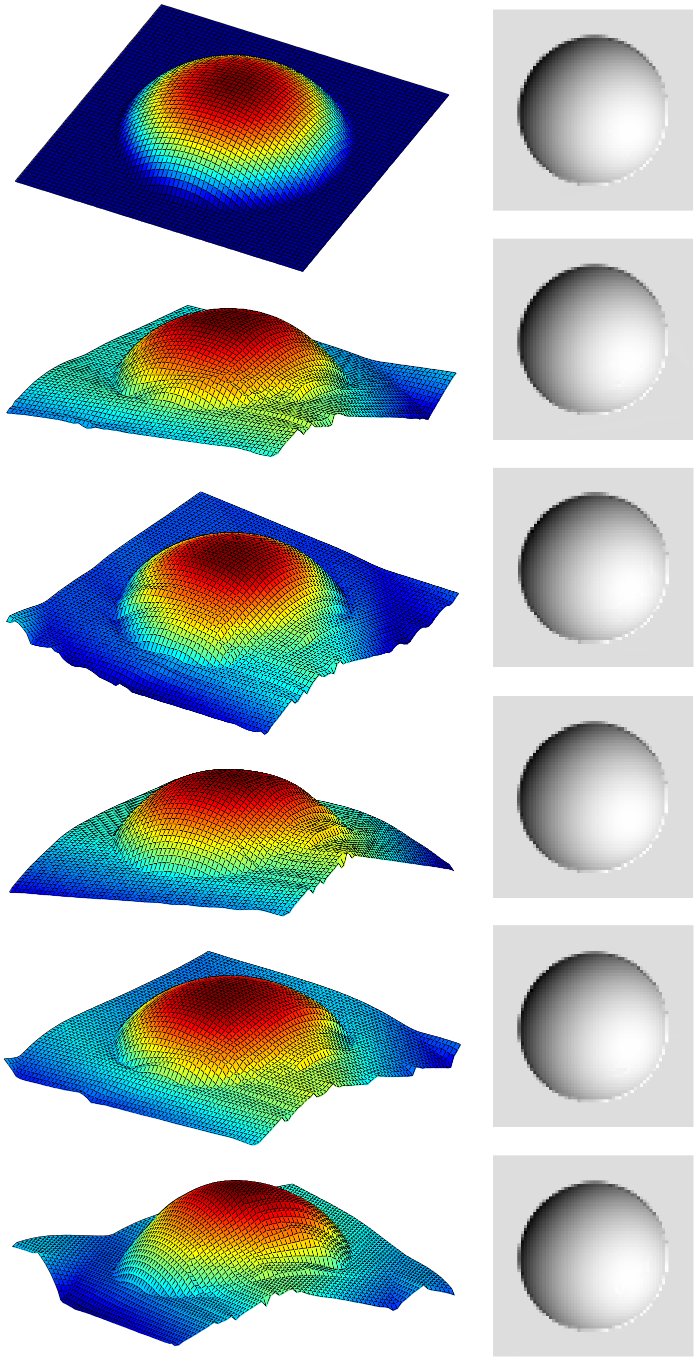


Figure 6.8: Artificially generated shading ambiguities of a Lambertian sphere cut by a plane (top left). The lower five surfaces, produced by our algorithm, generate very similar images when illuminated by the same light source $\mathbf{L} = (0.25, 0.433, 0.866)$.

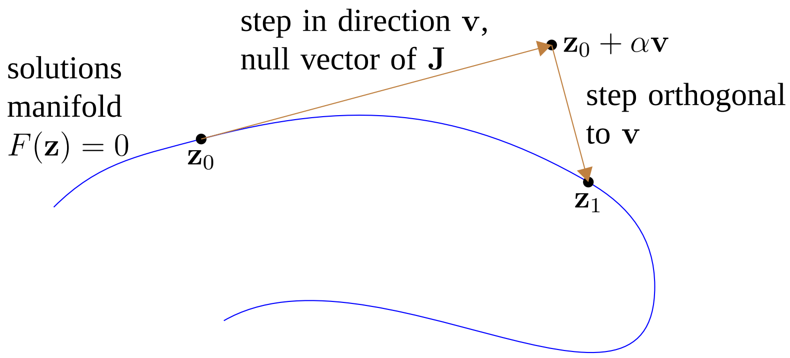


Figure 6.9: Generating a solution $\mathbf{z}_1$ from a known solution $\mathbf{z}_0$.

$\mathbf{z}_0$ and a light source $\mathbf{L}$. The idea is illustrated in figure 6.9. Starting from a solution $\mathbf{z}_0$ of the SFS equations, we take a large step that would change the image as little as possible. Consider the second order approximation $F(\mathbf{z}_0 + \mathbf{v}) \approx F(\mathbf{z}_0) + \nabla F(\mathbf{z}_0)^T\mathbf{v} + \frac{1}{2}\mathbf{v}^T\nabla^2 F(\mathbf{z}_0)\mathbf{v}$. Note that $F(\mathbf{z}_0) = 0$, $\nabla F(\mathbf{z}_0) = 0$. The Hessian at such a point is $\nabla^2 F(\mathbf{z}_0) = 2\mathbf{J}^T\mathbf{J}$. Hence, a desired direction $\mathbf{v}$ is a null vector of $\mathbf{J}$. Taking the step $\mathbf{z} = \mathbf{z}_0 + \alpha\mathbf{v}$, we get away from the solutions manifold. Applying our iterative procedure of section (6.3) with the constraint $\mathbf{v}^T(\mathbf{z} - (\mathbf{z}_0 + \alpha\mathbf{v})) = 0$ we get to a point $\mathbf{z}_1$ on the solutions manifold. This constraint ensures that we won't get close to $\mathbf{z}_0$. In our implementation, at each iteration we project the conjugate gradient search directions $\mathbf{d}$ so that $\mathbf{d} \cdot \mathbf{v} = 0$. A step size α is searched so that it is large and yet returning to the manifold is still possible.

It remains to describe how to compute the null vectors of $\mathbf{J}$. Note again that the matrix $\mathbf{J}$ is $MN \times (M+1)(N+1)$ and therefore large, sparse, and has at least $M + N + 1$ null vectors. This can be exploited in a divide-and-conquer scheme. We partition the image to four roughly equal blocks $I = \left[\begin{array}{c|c} I_1 & I_2 \\ \hline I_3 & I_4 \end{array}\right]$, and compute bases $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$ for the null spaces of $\mathbf{J}$ restricted to each block recursively (svd is used for small images). For a $\frac{M}{2} \times \frac{N}{2}$ subimage, we form its restricted Jacobian on a $(\frac{M}{2}+1) \times (\frac{N}{2}+1)$ extended grid, making it rank-deficient. Each column of the basis matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$ has $(\frac{M}{2}+1)(\frac{N}{2}+1)$ entries that correspond to heights on the extended subimages. The bases $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$ overlap on the middle row and column of I. We stitch these bases together to form a basis $\mathbf{B}$ for the null space of $\mathbf{J}$ by finding linear combinations of the null vectors that match on the overlapping pixels of I. Let $\hat{\mathbf{B}}_1$, $\hat{\mathbf{B}}_2$ be the rows of $\mathbf{B}_1, \mathbf{B}_2$ that correspond to their overlapping pixels along the middle column of I, and let $\begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}$ be the null space of $\begin{bmatrix} \hat{\mathbf{B}}_1 & -\hat{\mathbf{B}}_2 \end{bmatrix}$. Then $\mathbf{B}_1\mathbf{W}_1$ combined with $\mathbf{B}_2\mathbf{W}_2$ give a basis for the null space of $\mathbf{J}$ restricted to $\left[I_1 \; I_2\right]$. We combine $\mathbf{B}_3, \mathbf{B}_4$ and then the upper and lower parts in a similar way. Finally, we orthogonalize the resulting null space matrix $\mathbf{B}$.

It is convenient to sort the null space by smoothness from low to high frequency. We build a matrix $\mathbf{C}$ that measures smoothness by applying the filters $\left[1, -2, 1\right]$, $\left[1, -2, 1\right]^T$, $\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ at every pixel. Let $\mathbf{V}$ be the sorted right-singular vectors of $\mathbf{CB}$. Then the matrix $\mathbf{BV}$ contains the null vectors in sorted order. Some null vectors for a plane and a cylinder surfaces are shown in figures 6.6 (bottom) and 6.7.

6.6 Conclusions

In this chapter we applied tools specifically designed for polynomial systems to the SFS problem, both for polyhedral and curved surfaces. Their main advantage is in not requiring boundary conditions. In contrast, many propagation methods (e.g. [1,75,150]) rely on boundary conditions. Other propagation methods (e.g. [216]) rely on the existence of singular points (where $\mathbf{N} = \mathbf{L}$) in the image. While propagation approaches are fast and impressive reconstructions have been demonstrated, these reliances on the existence of initial information are conceptually unsatisfactory. In general, we cannot assume knowledge of curved boundaries, and wouldn't like the algorithm to completely break down when the singular points are occluded.

The SDP relaxation is a radically different approach to the SFS problem. It is interesting because convergence to a global minimum of the relaxed problem is guaranteed, without depending on the initial guess. However, the solution of the SDP relaxation does not lead directly to a solution of the original SFS problem.

Another method that does not depend on the initial guess is Pentland's linear SFS [146]. The difference is that under the Lambertian model, the true surface has a positive error in linear SFS, whereas the SDP relaxation satisfies equation (6.19) with zero slack. Of course, this does not imply that the SDP relaxation will find the optimal solution, since there might be solutions in the extended set of variables with lower smoothness or sum of matrix traces. On the other hand, if an SDP relaxation without smoothness does not have a solution, possibly because we added incompatible constraints, then we can conclude that some of our assumptions (e.g. light source direction) were wrong. This is impossible to infer from linear SFS, since the true surface has a positive error in the linear SFS model.

Additional depth constraints obtained from other depth cues or user interaction can be added easily to the system. These may include inequality constraints (e.g. (6.3) or front/behind relationships), which are difficult to incorporate into other methods.

While currently the results of the SFS relaxation are inferior to other SFS methods, it opens an entirely new direction for future research. At present SDP solvers limit the method to low-resolution images and low relaxation orders (note the possibility of variable relaxation orders at different pixels). It is conceivable that the method will do better with large-scale SDP solvers that will allow higher relaxation orders or incorporate smoothness terms over large windows. Better exploitation of sparseness, parallelism, and tighter approximations of the full SDP problem, may allow larger problems to be solved in the future. Other interesting directions for future research include improving the solution extraction from the moment matrices, and using the relaxation in a branch-and-bound scheme.

Ambiguities are a serious problem for all iterative SFS methods with no boundary conditions, since the algorithm can start converging to one solution in one part of the image and a conflicting solution in a remote part. For large and noisy images, this can occur even with boundary conditions. Therefore, some prior information is needed to choose between solutions. We proposed a new smoothness term for the iterative method, and used the sum of traces of the moment matrices in the SDP method. These priors are sensible, but obviously not a complete answer.

As a first step towards dealing with shading ambiguities, we introduced a method for generating ambiguous surfaces that produce the same image. This complements previous work on ambiguity of analytical surfaces, and observations about the convergence of some SFS algorithms to ambiguous surfaces. The ability to visualize the family of ambiguous surfaces may contribute to better understanding their extent.

Chapter 7

Conclusions

7.1 The big picture in low resolution

Single-image 3D surface reconstruction is a key problem in computer vision. While multi-view methods can provide depth estimates at some points, single-view cues provide only partial information and raise the problem of surface integration. The challenge in single-view reconstruction is both to identify the cues in the image and to solve the resulting optimization problems.

In this thesis we investigated optimization problems in single-image 3D reconstruction. We examined various types of constraints and formulations for shape from planar curves, shape from texture, and shape from shading. The constraints were categorized as linear, discrete or quadratic. Although we studied each case in isolation, they can be combined together. While there are numerous ways to perform the optimization, we recommend considering the following possibilities:

- If there is a significant number of linear constraints, solve the linear constraints first to obtain a low-dimensional linear subspace that contains (or close to) the solution.

- If the number of discrete-ambiguity constraints is small, do exhaustive search on them or on a small sample.

- If a reasonable starting guess is known (e.g. from a recognition module), form the problem as a quartic (or polynomial) minimization, and try an iterative solver with exact line search.

- One way to get an initial guess is to form an SDP relaxation from the discrete ambiguity constraints or the continuous quadratic constraints. In cases where multiple solutions are possible, a suitable regularization term or a symmetry-breaking term should be included. The relaxation can be used as an initial guess for an iterative procedure or local search refinement.

7.2 Summary of contributions

The main contributions of the thesis are summarized below.

In chapter 3

- We presented a unified analysis for the problem of shape from planar curves, including planar faces, planar cross sections, straight lines and parallel lines. The same scheme applies to the analysis of polyhedra, single-view modeling, structured light, and SHGCs. Previously these cases were treated separately and not in their full generality.

- We presented a characterization of the trivial subspace based on a geometric criterion. Under this criterion, the cost function is the same for both orthographic and perspective projections. It yields a more robust method compared to a direct least-squares solution. We showed empirically that for randomly oriented planes it makes sense to pick a solution orthogonal to the basic trivial subspace.

In chapter 4

- We formulated the problem of inferring a smooth surface from texture cues as a discrete decision problem.

- We applied an SDP relaxation with various rounding heuristics to this problem. Although the problem is NP-hard, very reasonable approximate solutions can be obtained, even for instances with over 1,000 decision variables.

In chapter 6

- We applied polynomial-solving techniques to the SFS problem.
  - By using a homotopy solver, we showed that exact solutions can be found at a vertex of a generic polyhedron, and in generic cases this can be extended to a solution over the entire polyhedron. However, the approach is sensitive to noise.
  - We demonstrated the use of exact line search for the SFS problem. The technique is widely known in the optimization community, yet hardly ever used in computer vision. It easily applies to many other problems, such as single-view modeling from planarity and orthogonality constraints.
  - We implemented a sparse SDP relaxation for the SFS problem. While currently the results are inferior to other SFS algorithms, this method does not depend on an initial guess, and opens a new direction for future research.
- We proposed a method for generating artificial shading ambiguities. This method will allow researchers to test surface priors on different surfaces that produce the same image.

Bibliography

[1] A.H. Ahmed and A.A. Farag. A new formulation for shape from shading for non-lambertian surfaces. In *Proc. CVPR'06*, pages 1817–1824, 2006. 85, 111

[2] S. Arora, E. Berger, E. Hazan, G. Kindler, and M. Safra. On non-approximability for quadratic programs. In *Proc. FOCS'05*, pages 206–215, 2005. 64

[3] J.J. Atick, P.A. Griffin, and A.N. Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996. 6, 75

[4] X. Bai, H. Yu, and E.R. Hancock. Graph matching using spectral embedding and semidefinite programming. In *BMVC'04*, pages 297–307, 2004. 11

[5] I. Barnes and K. Zhang. Instability of the eikonal equation and shape from shading. *Math. Model. Numer. Anal.*, 34(1):127–138, 2000. 106

[6] R. Bauer. Distribution of points on a sphere with application to star catalogs. *Journal of guidance, control and dynamics*, 23(1):130–137, 2000. 104

[7] P.N. Belhumeur, D.J. Kriegman, and A.L. Yuille. The bas-relief ambiguity. *IJCV*, 35(1):33–44, 1999. 20, 84, 106

[8] B.M. Bennett, D.D. Hoffman, J.E. Nicola, and C. Prakash. Inferring structure from motion: A homotopy algorithm. In *Proc. Workshop on Visual Motion*, pages 238–245, 1989. 92

[9] S.J. Benson and Y. Ye. Algorithm 875: DSDP5: Software for semidefinite programming. *ACM Trans. Math. Software*, 34(3), 2008. 67, 104

[10] A. Bhargava and S.R. Kosaraju. Derandomization of dimensionality reduction and sdp based algorithms. In *Proc. WADS'05*, pages 396–408, 2005. 65

[11] M. Bichsel and A.P. Pentland. A simple algorithm for shape for shading. In *Proc. CVPR'92*, pages 459–465, 1992. 85

[12] P. Biswas, H. Aghajan, and Y. Ye. Semidefinite programming algorithms for sensor network localization using angle information. In *Asilomar Conference on Signals, Systems and Computers*, pages 220–224, 2005. 102

[13] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006. 102

[14] J.-Y. Bouguet, M. Weber, and P. Perona. What do planar shadows tell us about scene geometry? In *Proc. CVPR'99*, pages 514–520, 1999. 14, 17, 20, 25, 28, 31, 40, 44

[15] H. Boyaci and L.T. Maloney. Binocular perception of shape from shading/contour is invariant under ordinal transformations of image intensities. *Journal of Vision*, 2(7):309a, 2002. 88

[16] P. Breton and S.W. Zucker. Shadows and shading flow fields. In *Proc. CVPR'96*, pages 782–789, 1996. 90

[17] M.J. Brooks, W. Chojnacki, and R. Kozera. Impossible and ambiguous shading patterns. *IJCV*, 7(2):119–126, January 1992. 83

[18] S. Burer, R.D.C. Monteiro, and Y. Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521, 2002. 11, 66, 76

[19] J.B. Burns, R.S. Weiss, and E.M. Riseman. The non-existence of general-case view-invariants. pages 120–131, 1992. 17

[20] M. Byröd, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *IJCV*, 84(3):237–256, 2009. 92

[21] Y. Caspi and M. Werman. Vertical parallax from moving shadows. In *Proc. CVPR'06*, pages 2309–2315, 2006. 40, 44

[22] T.F. Chan, J.R. Gilbert, and S.-H. Teng. Geometric spectral partitioning. Technical Report Tech. Report CSL-94-15, Xerox PARC, 1995. 66

[23] M. Chandraker, S. Agarwal, D. Kriegman, and S. Belongie. Globally optimal algorithms for stratified autocalibration. *IJCV*. 11

[24] M. Chandraker and D. Kriegman. Globally optimal bilinear programming for computer vision applications. In *Proc. CVPR'08*, pages 1–8, 2008. 11

[25] J.Y. Chang, K.M. Lee, and S.U. Lee. Shape from shading using graph cuts. *Pattern Recognition*, 41(12):3749–3757, 2008. 56

[26] D. Chen, W. Gao, and X. Chen. A new approach of recovering 3-D shape from structure-lighting. In *Proc. ICSP'96*, pages 839–842, 1996. 40

[27] H.F. Chen, P.N. Belhumeur, and D.W. Jacobs. In search of illumination invariants. In *Proc. CVPR'00*, pages 254–261, 2000. 4

[28] X. Chen, S.B. Kang, Y.-Q. Xu, J. Dorsey, and H.-Y. Shum. Sketching reality: Realistic interpretation of architectural designs. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(2):1–15, 2008. 7

[29] W. Chojnacki, M.J. Brooks, A. van den Hengel, and D. Gawley. From FNS to HEIV: A link between two vision parameter estimation methods. *PAMI*, 26(2):264–268, 2004. 46

[30] M.B. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971. 13

[31] C. Colombo, A. Del Bimbo, and F. Pernici. Metric 3D reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. *PAMI*, 27(1):99–114, 2005. 14, 47

[32] P. Company, A. Piquer, M. Contero, and F. Naya. A survey on geometrical reconstruction as a core technology to sketch-based modeling. *Computers & Graphics*, 29(6):892–904, 2005. 5

[33] M. Cooper. *Line Drawing Interpretation*. Springer, 2008. 13

[34] H. Crapo. Invariant-theoretic methods in scene analysis and structural mechanics. *Journal of Symbolic Computation*, 11(5-6):523–548, 1991. 24

[35] E. de Klerk, D.V. Pasechnik, and J.P. Warners. On approximate graph colouring and max-k-cut algorithms based on the ϑ-function. *J. Comb. Optim.*, 8(3):267–294, 2004. 70

[36] P. Deift and J. Sylvester. Some remarks on the shape-from-shading problem in computer vision. *J. Math. Anal. Appl.*, 84(1):235–248, 1981. 84

[37] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, first edition, 1973. 12

[38] J.-D. Durou and D. Piau. Ambiguous shape from shading with critical points. *Journal of Mathematical Imaging and Vision*, 12(2):99–108, 2000. 106

[39] A. Ecker and A.D. Jepson. Polynomial shape from shading. In *Proc. CVPR'10*, 2010. 2

[40] A. Ecker, A.D. Jepson, and K.N. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. In *Proc. ECCV'08*, volume 1, pages 127–140, 2008. 2

[41] A. Ecker, K.N. Kutulakos, and A.D. Jepson. Shape from planar curves: A linear escape from flatland. In *Proc. CVPR'07*, pages 1–8, 2007. 2, 25

[42] G. Falk. Interpretation of imperfect line data as a three-dimensional scene. *Artificial Intelligence*, 3:101–144, 1972. 14, 19

[43] O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In *Proc. CVPR'92*, pages 321–334, 1992. 92

[44] M. Fazel, H. Hindi, and S.P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proc. American Control Conference*, volume 3, pages 2156–2162, 2003. 104

[45] C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proc. 19th Design Automation Conference*, pages 175–181, 1982. 67

[46] P.-O. Fjällström. Algorithms for graph partitioning: A survey. *Linkoping Electronic Articles in Computer and Information Science*, 3(10), 1998. 66

[47] R.W. Fleming, A. Torralba, and E.H. Adelson. Specular reflections and the perception of shape. *Journal of Vision*, 4(9):798–820, 2004. 90

[48] D.A. Forsyth. Shape from texture and integrability. In *Proc. ICCV'01*, pages 447–452, 2001. 59, 68

[49] D.A. Forsyth. Shape from texture without boundaries. In *Proc. ECCV'02*, pages 225–239, 2002. 59, 77

[50] W.T. Freeman. Exploiting the generic viewpoint assumption. *IJCV*, 20(3):243–261, 1996. 106

[51] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000. 7

[52] A. Frieze and M. Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81, 1997. 70

[53] R. Furukawa and H. Kawasaki. Geometrical constraint based 3D reconstruction using implicit coplanarities. In *BMVC'07*, pages 780–789, 2007. 25

[54] R. Furukawa and H. Kawasaki. Laser range scanner based on self-calibration techniques using coplanarities and metric constraints. *CVIU*, 113(11):1118–1129, November 2009. 25

[55] E.R. Gansner, Y. Koren, and S.C. North. Graph drawing by stress majorization. In *Proc. 12th International Symposium on Graph Drawing*, pages 239–250, 2004. 77

[56] J. Gårding. Shape from texture for smooth curved surfaces in perspective projection. *Journal of Mathematical Imaging and Vision*, 2:329–352, 1992. 75

[57] D. Gelli and D. Vitulano. Speed up of shape from shading using graduated non convexity. In *Proc. DGCI'03*, pages 504–513, 2003. 95

[58] A. Gilchrist. *Seeing Black and White*. Oxford University Press, 2006. 86

[59] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. 11, 58, 65

[60] C. Gotsman and S. Toledo. On the computation of null spaces of sparse rectangular matrices. *SIAM J. Matrix Anal. Appl.*, 30(2):445–463, 2008. 10

[61] K. Grill-Spector. What has fMRI taught us about object recognition? In *Object Categorization: Computer and Human Vision Perspectives*, pages 102–128. Cambridge University Press, 2009. 3

[62] I.J. Grimstead and R.R. Martin. Creating solid models from single 2D sketches. In *Proc. 3rd ACM Symposium on Solid Modeling and Applications*, pages 323–337, 1995. 28

[63] A. Gültekin and M. Gökmen. Adaptive shape from shading. *Turkish Journal of ELEKTRIK*, 6(2):61–73, 1998. 99

[64] A. Guzman-Arenas. *Computer Recognition of Three-Dimensional Objects in a Visual Scene*. PhD thesis, MIT, 1968. 12

[65] K.M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970. 31

[66] F. Han and S.-C. Zhu. A two-level generative model for cloth representation and shape from shading. *PAMI*, 29(7):1230–1243, 2007. 7

[67] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *Beyond Patches Workshop, CVPR'06*, page 15, 2006. 7

[68] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZIB-Report ZR-00-34, TU Berlin, 2000. 11

[69] D. Henrion and J.B. Lasserre. Detecting global optimality and extracting solutions in GloptiPoly. In *Positive Polynomials in Control*, pages 293–310. Springer, 2005. 104

[70] C. Hernández, G. Vogiatzis, and R. Cipolla. Shadows in three-source photometric stereo. In *Proc. ECCV'08*, pages 290–303, 2008. 61

[71] A. Heyden. *Geometry and algebra of multiple projective transformations*. PhD thesis, Lund Institute of Technology, 1995. 14

[72] A. Heyden. On the consistency of line-drawings, obtained by projections of piecewise planar objects. *Journal of Mathematical Imaging and Vision*, 6(4):393–412, 1996. 23, 28

[73] D. Hoiem, A.A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 24(3):577–584, 2005. 7

[74] B.K.P. Horn. Understanding image intensities. *Artificial Intelligence*, 8(2):201–231, 1977. 95

[75] B.K.P. Horn and M.J. Brooks, editors. *Shape from Shading*. MIT Press, 1989. 3, 7, 81, 85, 87, 90, 91, 92, 98, 111

[76] B.K.P. Horn, R.S. Szeliski, and A.L. Yuille. Impossible shaded images. *PAMI*, 15(2):166–170, February 1993. 83

[77] R. Huang and W.A.P. Smith. A shape-from-shading framework for satisfying data-closeness and structure-preserving smoothness constraints. In *Proc. BMVC'09*, 2009. 99

[78] X. Huang, J. Gao, L. Wang, and R. Yang. Examplar-based shape from shading. In *Proc. 3DIM'07*, pages 349–356, 2007. 7

[79] D.A. Huffman. Impossible objects as non-sense sentences. *Machine Intelligence*, 6:295–323, 1971. 13

[80] N.E. Hurt. Mathematical methods in shape-from-shading: A review of recent results. *Acta Applicandae Mathematicae*, 23(2):163–188, 1991. 81

[81] H.W. Jensen, S.R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *Proc. SIGGRAPH'01*, pages 511–518, 2001. 91

[82] D.R. Jones, C.D. Perttunen, and B.E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993. 95

[83] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *IJCV*, 74(1):3–15, 2007. 92

[84] K. Kanatani. Constraints on length and angle. *CVGIP*, 41(1):28–42, 1988. 9

[85] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993. 75

[86] K. Kanatani. Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis. *IJCV*, 80(2):167–188, 2008. 46

[87] C.-Y. Kao, S. Osher, and J. Qian. Lax-Friedrichs sweeping scheme for static hamilton-jacobi equations. *Journal of Computational Physics*, 196(1):367–391, 2004. 85

[88] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998. 67

[89] H. Kawasaki and R. Furukawa. Dense 3D reconstruction method using coplanarities and metric constraints for line laser scanning. In *Proc. 3DIM'07*, pages 149–158, 2007. 25, 48

[90] H. Kawasaki and R. Furukawa. Shape reconstruction from cast shadows using coplanarities and metric constraints. In *ACCV'07*, volume 2, pages 847–857, 2007. 25

[91] H. Kawasaki and R. Furukawa. Shape reconstruction and camera self-calibration using cast shadows and scene geometries. *IJCV*, 83(2):135–148, 2009. 25, 46

[92] I. Kemelmacher-Shlizerman, R. Basri, and B. Nadler. 3D shape reconstruction of mooney faces. In *Proc. CVPR'08*, pages 1–8, 2008. 106

[93] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970. 66, 67

[94] J. Keuchel. Multiclass image labeling with semidefinite programming. In *Proc. ECCV'06*, pages 454–467, 2006. 11

[95] J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *PAMI*, 25(11):1364–1379, 2003. 11

[96] R. Kimmel and J.A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(3):237–244, 2001. 85

[97] L. Kirousis and C.H. Papadimtriou. The complexity of recognizing polyhedral scenes. *J. of Computer and System Sciences*, 37(1):14–38, 1988. 13, 14, 76

[98] D.A. Kleffner and V.S. Ramachandran. On the perception of shape from shading. *Perception & Psychophysics*, 52:18–36, 1992. 86, 87

[99] A.V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001. 10

[100] J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.

[101] J.J. Koenderink and S.C. Pont. Irradiation direction from texture. *JOSA A*, 20(10):1875–1882, 2003. 85

[102] J.J. Koenderink and A.J. van Doorn. Affine structure from motion. *JOSA-A*, 8(2):377–385, 1991. 21

[103] J.J Koenderink and A.J. van Doorn. Relief: pictorial and otherwise. *Image and Vision Computing*, 13(5):321–334, 1995. 87

[104] J.J. Koenderink and A.J. van Doorn. Shape from Chebyshev nets. In *Proc. ECCV'98*, pages 215–225, 1998. 62

[105] J.J. Koenderink, A.J. van Doorn, and A.M.L. Kappers. Surface perception in pictures. *Perception & Psychophysics*, 52(5):487–496, 1992. 87

[106] J.J. Koenderink, A.J. van Doorn, A.M.L. Kappers, and J.T. Todd. Ambiguity and the 'mental eye' in pictorial relief. *Perception*, 30(4):431–448, 2001. 3, 21

[107] Y. Koren, L. Carmel, and D. Harel. Drawing huge graphs by algebraic multigrid optimization. *Multiscale Modeling and Simulation*, 1(4):645–673, 2003. 31

[108] R. Kozera. Uniqueness in shape from shading revisited. *JMIV*, 7(2):123–138, 1997. 84, 93, 106

[109] D.J. Kriegman and P.N. Belhumeur. What shadows reveal about object structure. *JOSA A*, 18(8):1804–1813, 2001. 20

[110] K. Krishnan and J.E. Mitchell. A semidefinite programming based polyhedral cut and price approach for the maxcut problem. *Comp. Optim. Appl.*, 33(1):51–71, 2006. 67

[111] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. ICCV'88*, pages 238–249, 1988. 17

[112] M.S. Langer and H.H. Bülthoff. Perception of shape from shading on a cloudy day. Technical Report 73, Max Planck Institute for Biological Cybernetics, 1999. 3, 87

[113] J.B. Lasserre. Convergent sdp-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006. 104

[114] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging Applications of Algebraic Geometry*, volume 149 of *IMA Volumes in Mathematics and its Applications*, pages 157–270. Springer, 2009. 102

[115] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In *Handbook on Discrete Optimization*, pages 393–514. Elsevier, 2005. 11

[116] K.M. Lee and C.C.J. Kuo. Shape from shading with a linear triangular element surface model. *PAMI*, 15(8):815–822, 1993. 95

[117] K.M. Lee and C.C.J. Kuo. Shape from shading with a generalized reflectance map model. *CVIU*, 67(2):143–160, 1997. 85

[118] T.L. Lee, T.Y. Li, and C.H. Tsai. HOM4PS-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2-3):109–133, 2008. 96

[119] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3D object from a single freehand line drawing. *Computer-Aided Design*, 28(8):651–663, 1996. 16, 23, 47

[120] H. Lipson and M. Shpitalni. Correlation-based reconstruction of a 3D object from a single freehand sketch. In *Proc. AAAI Spring Symposium on Sketch Understanding*, pages 99–104, 2002. 7

[121] J. Liu, L. Cao, Z. Li, and X. Tang. Plane-based optimization for 3D object reconstruction from single line drawings. *PAMI*, 30(2):315–327, 2008. 48

[122] J. Liu, Y.T. Lee, and W.-K. Cham. Identifying faces in a 2D line drawing representing a manifold object. *PAMI*, 24(12):1579–1593, 2002. 13

[123] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD'04*, pages 284–289, 2004. 104

[124] Z.-Q. Luo and S. Zhang. A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints. *SIAM J. on Optimization*, 20(4):1716–1736, 2010. 99

[125] J. Nie M. Mevissen, M. Kojima and N. Takayama. Solving partial differential equations via sparse sdp relaxations. *Pacific Journal of Optimization*, 4(2):213–241, 2008. 102, 104

[126] J. Malik. Interpreting line drawings of curved objects. *IJCV*, 1(1):73–103, 1987. 13

[127] P. Mamassian and R. Goutcher. Prior knowledge on the illumination position. *Cognition*, 81(1):B1–B9, 2001. 88

[128] P. Mamassian, I. Jentzsch, B.A. Bacon, and S.R. Schweinberger. Neural correlates of shape from shading. *NeuroReport*, 14(7):971–975, 2003. 86

[129] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman and Company, 1982. 3, 7

[130] G. Medioni. Generic object recognition by inference of 3-D volumetric parts. In *Object Categorization: Computer and Human Vision Perspectives*, pages 87–101. Cambridge University Press, 2009. 1

[131] R.D.C. Monteiro. First- and second-order methods for semidefinite programming. *Mathematical Programming*, B97:209–244, 2003. 76

[132] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed form. In *Proc. CVPR'09*, 2009. 92

[133] S. Naito and A. Rosenfeld. Shape from random planar features. *CVGIP*, 42(3):345–370, 1988. 62, 67

[134] K. Nakayama, Z.J. He, and S. Shimojo. Visual surface representation: A critical link between lower-level and higher level vision. In *An Invitation to Cognitive Science*, volume 2, pages 1–70. MIT Press, second edition, 1995. 3

[135] D. Nandy and J. Ben-Arie. Shape from recognition: A novel approach for 3-D face shape recovery. *IEEE Transactions on Image Processing*, 10(2):206–217, 2001. 7

[136] T. Okatani and K. Deguchi. Closed form solution of local shape from shading at critical points. *IJCV*, 40(2):169–178, 2000. 83

[137] J. Oliensis. Uniqueness in shape from shading. *IJCV*, 6(2):75–104, 1991. 83

[138] J. Oliensis and P. Dupuis. Direct method for reconstructing shape from shading. *SPIE*, 1570:116–128, 1991. 85

[139] L. Olsen, F.F. Samavati, M.C. Sousa, and J.A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009. 5

[140] C. Olsson, A.P. Eriksson, and F. Kahl. Improved spectral relaxation methods for binary quadratic optimization problems. *CVIU*, 112(1):3–13, 2008. 11, 76

[141] R. Onn and A. Bruckstein. Integrability disambiguates surface recovery in two-image photometric stereo. *IJCV*, 5(1):105–113, 1990. 61

[142] S.E. Palmer. *Vision Science: Photons to Phenomenology*. MIT Press, 1999. 3, 90

[143] P. Parodi, R. Lancewicki, A. Vijh, and J.K. Tsotsos. Empirically-derived estimates of the complexity of labeling line drawings of polyhedral scenes. *Artificial Intelligence*, 105(1–2):47–75, 1998. 13

[144] P. Parodi and G. Piccioli. 3D shape reconstruction by using vanishing points. *PAMI*, 18(2):211–217, 1996. 20

[145] M.A. Penna. A shape from shading analysis for a single perspective image of a polyhedron. *PAMI*, 11(6):545–554, 1989. 85, 92, 95

[146] A.P. Pentland. Linear shape from shading. *IJCV*, 4(2):153–162, March 1990. 112

[147] R.Z. Ping-Sing, R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape from shading: A survey. *PAMI*, 21(8):690–706, 1999. 81, 84, 92, 97, 98, 101

[148] Z. Pizlo, Y. Li, and G. Francis. A new look at binocular stereopsis. *Vision Research*, 45(17):2244–2255, 2005. 16

[149] E. Prados. *Application of the theory of the viscosity solutions to the Shape From Shading problem*. PhD thesis, INRIA Sophia Antipolis, 2004. 81, 85

[150] E. Prados and O. Faugeras. A generic and provably convergent shape-from-shading method for orthographic and pinhole cameras. *IJCV*, 65(1-2):97–125, 2005. 85, 111

[151] L. Qi, Z. Wan, and Y.-F. Yang. Global minimization of normal quartic polynomials based on global descent directions. *SIAM Journal on Optimization*, 15(1):275–302, 2005. 99

[152] V. Rabaud and S. Belongie. Linear embeddings in non-rigid structure from motion. In *Proc. CVPR'09*, pages 2427–2434, 2009. 92

[153] V.S. Ramachandran. Perceiving shape from shading. *Scientific American*, 252(2):76–83, 1988. 86

[154] T.H. Reiss. *Recognizing Planar Objects Using Invariant Image Features*. Springer-Verlag, 1993. 17

[155] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Technical report, Alpen-Adria-Universität Klagenfurt, Inst. f. Mathematik, 2007. 11, 76

[156] I. Rigoutsos. Homotopies: A panacea or just another method? Technical Report UR-CSD-TR-201, University of Rochester, 1986. 95

[157] L.G. Roberts. *Machine Perception of Three-Dimensional Solids*. PhD thesis, MIT, 1963. 3, 6, 12

[158] L. Ros. *A Kinematic-Geometric Approach to Spatial Interpretation of Line Drawings*. PhD thesis, Technical University of Catalonia, 2000. 12, 14, 24

[159] L. Ros and F. Thomas. Numerical analysis of the instantaneous motions of panel-and-hinge frameworks and its application to computer vision. In *Proc. of the 2nd Workshop on Computational Kinematics*, pages 199–210, 2001. 24, 28

[160] L. Ros and F. Thomas. Overcoming superstrictness in line drawing interpretation. *PAMI*, 24(4):456–466, 2002. 27, 28

[161] C. Rothwell and J. Stern. Understanding the shape properties of trihedral polyhedra. Technical Report 2661, INRIA, 1995. 16, 17, 26, 31

[162] C.A. Rothwell. *Object Recognition through Invariant Indexing*. Oxford University Press, 1995. 16, 17

[163] C.A. Rothwell, D.A. Forsyth, A. Zisserman, and J.L. Mundy. Extracting projective structure from single perspective views of 3D point sets. In *Proc. ICCV'93*, pages 573–582, 1993. 16, 17

[164] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992. 85

[165] R. Sagawa, Y. Ota, Y.i. Yagi, R. Furukawa, N. Asada, and H. Kawasaki. Dense 3D reconstruction method using a single pattern for fast moving object. In *Proc. ICCV'2009*, pages 1779–1786, 2009. 44

[166] R. Sara. *Local Shading Analysis via Isophotes Properties*. PhD thesis, Johannes Kepler University Linz, 1994. 90

[167] B.V.H. Saxberg. *A Modern Differential Geometric Approach to Shape from Shading*. PhD thesis, MIT, 1989. 83

[168] A. Saxena, M. Sun, and A.Y. Ng. Make3D: Learning 3D scene structure from a single still image. *PAMI*, 31(5):824–840, 2009. 7

[169] C. Schellewald and C. Schnörr. Probabilistic subgraph matching based on convex relaxation. In *Proc. EMMCVPR'05*, pages 171–186, 2005. 11

[170] R.A. Shatas, J.F. Marshall, and M.A. Pomerantz. Dependence of secondary electron emission upon angle of incidence of 1.3-Mev primaries. *Physical Review*, 102(3):682–686, 1956. 91

[171] R.N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 3

[172] H. Shimodaira. A shape-from-shading method of polyhedral objects using prior information. *PAMI*, 28(4):612–624, 2006. 47, 95

[173] I. Shimshoni and J. Ponce. Recovering the shape of polyhedra using line-drawing analysis and complex reflectance models. *CVIU*, 65(2):296–310, 1997. 27, 47

[174] M. Shpitalni and H. Lipson. Identification of faces in a 2D line drawing projection of a wireframe object. *PAMI*, 18(10):1000–1012, 1996. 13

[175] H.-Y. Shum, R. Szeliski, S. Baker, M. Han, and P. Anandan. Interactive 3D modeling from multiple images using scene regularities. In *Proc. SMILE'98, European Workshop on 3D Structure from Multiple Images of Large-Scale Environments*, pages 236–252, 1998. 20

[176] P. Sinha and E. Adelson. Verifying the "consistency" of shading patterns and 3-D structures. In *Proc. IEEE Workshop on Qualitative Vision*, pages 71–80, 1993. 98

[177] P. Sinha and E.H. Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *Proc. ICCV'93*, pages 156–163, 1993.

[178] G.B. Smith. The relationship between image irradiance and surface orientation. In *Proc. CVPR'83*, pages 14–19, 1983. 85

[179] A.J. Sommese and C.W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005. 95

[180] G. Sparr. Depth computations from polyhedral images. *Image and Vision Computing*, 10(10):683–688, 1992. 23

[181] K.A. Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17(1–3):47–73, 1981. 16, 51, 53

[182] P.F. Sturm and S.J. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single images. In *Proc. BMVC'99*, pages 265–274, 1999. 20, 28, 36

[183] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, 1986. 12, 14, 22, 25, 27, 34, 47, 95

[184] J. Sun and P. Perona. Where is the sun? *Nature Neuroscience*, 1(3):183–184, 1998. 88

[185] R. Szeliski. Fast shape from shading. In *Proc. ECCV'90*, pages 359–368, 1990. 95, 98

[186] A. Tankus, N. Sochen, and Y. Yeshurun. Shape-from-shading under perspective projection. *IJCV*, 63(1):21–43, 2005. 85

[187] J. Tessendorf. Simulating ocean surface. SIGGRAPH'04 course notes, 2004. 90

[188] J.T. Todd and F.D. Reichel. Visual perception of smoothly curved surfaces from double-projected contour patterns. *Journal of Experimental Psychology: Human Perception and Performance*, 16(3):665–674, 1990. 16, 48, 51, 53

[189] M.J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001. 11

[190] P.H.S. Torr. Solving Markov random fields using semi definite programming. In *Proc. Ninth International Workshop on Artificial Intelligence and Statistics*, 2003. 11

[191] D.A. Trytten and M. Tuceryan. The construction of labeled line-drawings from intensity images. *Pattern Recognition*, 28(2):171–198, 1995. 13

[192] P.U. Tse. A contour propagation account of surface filling-in and volume formation. *Psychological Review*, 109(1):91–115, 2002. 25

[193] C.W. Tyler. Spatial form as inherently three dimensional. In *Seeing Spatial Form*, pages 67–88. Oxford University Press, 2006. 3

[194] S. Ullman. The interpretation of structure from motion. *Proc. of the Royal Society of London*, B-203:405–426, 1979. 55

[195] F. Ulupinar and R. Nevatia. Perception of 3-D surfaces from 2-D contours. *PAMI*, 15(1):3–18, 1993. 28

[196] F. Ulupinar and R. Nevatia. Shape from contour: Straight homogeneous generalized cylinders and constant cross section generalized cylinders. *PAMI*, 17(2):120–135, 1995. 14, 25

[197] P.A.C. Varley. *Automatic Creation of Boundary-Representation Models from Single Line Drawings*. PhD thesis, Cardiff University, 2003. 12, 13, 14

[198] P.A.C. Varley and R.R. Martin. Estimating depth from line drawings. In *Proc. 7th ACM Symposium on Solid Modeling and Applications*, pages 180–191, 2002. 9, 28, 47

[199] P.A.C. Varley, R.R. Martin, and H. Suzuki. Frontal geometry from sketches of engineering objects: is line labelling necessary? *Computer-Aided Design*, 37(12):1285–1307, 2005. 14

[200] P.W. Verbeek and B.J.H. Verwer. Shading from shape, the eikonal equation solved by grey-weighted distance transform. *Pattern Recognition Letters*, 11(10):681–690, 1990. 85

[201] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006. 102, 104

[202] D. Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, pages 19–91. McGraw-Hill, 1975. 13

[203] L.T. Watson. Globally convergent homotopy methods: A tutorial. *App. Math. and Comp.*, 31:369–396, 1989. 95

[204] K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *IJCV*, 70(1):77–90, 2006. 92, 99

[205] K.Q. Weinberger, F. Sha, Q. Zhu, and L.K. Saul. Graph laplacian regularization for large-scale semidefinite programming. In *NIPS*, pages 1489–1496, 2006. 77

[206] D. Weinshall. Local shape approximation from shading. *Journal of Mathematical Imaging and Vision*, 4(2):119–138, 1994. 90

[207] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein. Digital bas-relief from 3D scenes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007. 20, 106

[208] R. White and D.A. Forsyth. Combining cues: Shape from shading and texture. In *Proc. CVPR'06*, pages 1809–1816, 2006. 75

[209] L.B. Wolff. A photometric invariant and shape constraints at parabolic points. In *Proc. ICCV'90*, pages 344–349, 1990. 90

[210] K.C. Wong and J. Kittler. Recognition of polyhedral objects using triplets of projected spatial edges based on a single perspective image. *Pattern Recognition*, 34(3):561–586, 2001. 9

[211] P.L. Worthington and E.R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *PAMI*, 21(12):1250–1267, 1999. 84

[212] Y. Wu, S.S. Iyengar, R. Jain, and S. Bose. A new generalized computational framework for finding object orientation using perspective trihedral angle constraint. *PAMI*, 16(10):961–975, 1994. 9

[213] J. Yang, N. Ohnishi, D. Zhang, and N. Sugie. Determining a polyhedral shape using interreflections. In *Proc. CVPR'97*, pages 110–115, 1997. 95

[214] E. Yarack and J. Carletta. An evaluation of move-based multi-way partitioning algorithms. In *Proc. ICCD'00*, pages 363–369, 2000. 72

[215] H. Yu and E.R. Hancock. Graph seriation using semi-definite programming. In *Proc. GbRPR'05*, pages 63–71, 2005. 11

[216] S.Y. Yuen, Y.Y. Tsui, and C.K. Chow. A fast marching formulation of perspective shape from shading under frontal illumination. *Pattern Recognition Letters*, 28(7):806–824, 2007. 85, 111

[217] L. Zhang, G. Dugas-Phocion, J.-S. Samson, and S.M. Seitz. Single view modeling of free-form scenes. In *Proc. CVPR'01*, pages 990–997, 2001. 6, 75

[218] Q. Zhu and J. Shi. Shape from shading: Recognizing the mountains through a global view. In *Proc. CVPR'06*, pages 1839–1846, 2006. 56, 75, 76, 85