Reconstructing Geometry from Its Latent Structures

A Thesis

Submitted to the Faculty

of

Drexel University

by

Geoffrey Oxholm

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

June 2014

© Copyright 2014
Geoffrey Oxholm. All Rights Reserved.

Dedications

I dedicate this thesis to science.

Acknowledgments

I wish to express deep gratitude to my adviser, Dr. Ko Nishino. It was his belief in me that gave me the strength to step far outside of my comfort zone. His demand for excellence, and his understanding of the personal and professional challenges associated with the pursuit of a Ph.D. are unmatched.

My supportive and encouraging parents have given me opportunities that much of the world cannot imagine. Enabling me to take risks, think creatively, and explore my possibilities, they are responsible for the most rewarding and valuable experiences of my life, including pursuing this Ph.D. and meeting my wife. I am constantly humbled by the privileges afforded to me in large part through their hard work, and open-mindedness.

I would not have completed this chapter of my life if it were not for the encouragement and support from my wife, Pita. During the most challenging deadlines it was an immeasurable source of comfort to know that she would be there when I eventually returned home from the lab, or would come visit me, if I decided to work all night.

I will forever think of my lab-mates as my research family; especially Lou, Prabin, Steve, Gabe, Linge and Manolya. It is hard to imagine how valuable a quick game of darts, or a beer, or even just a chat over lunch can be until you really need one.

Finally, I wish to acknowledge the taxpayer, who allowed me to take this time to pursue topics whose value they will likely never directly know. Specifically, my work was supported by the National Science Foundation awards NSF IIS-0746717, IIS-0803670, IIS-0964420, IIS-1353235, and the Office of Naval Research grants N00014-11-1-0099 and N00014-14-1-0316; all funded by the people who live in this country (whether documented or not). Knowing the source of my funding constantly served as an inspiration for me, and I hope that I can continue to repay this incredible opportunity through my service as an engaged citizen.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Reconstructing Geometry from Its Latent Structures
Geoffrey Oxholm
Ko Nishino, Ph.D.

Our world is full of objects with complex shapes and structures. Through extensive experience, humans quickly develop an intuition about how objects are shaped, and what their material properties are simply by analyzing their appearance. We engage this intuitive understanding of geometry in nearly everything we do.

It is not surprising then, that a careful treatment of geometry stands to give machines a powerful advantage in the many tasks of visual perception. To that end, this thesis focuses on geometry recovery in a wide range of real-world problems. First, we describe a new approach to image registration. We observe that the structure of the imaged subject becomes embedded in the image intensities. By minimizing the change in shape of these intensity structures we ensure a physically realizable deformation. We then describe a method for reassembling fragmented, thin-shelled objects from range-images of their fragments using only the geometric and photometric structure embedded in the boundary of each fragment. Third, we describe a method for recovering and representing the shape of a geometric texture (such as bark, or sandpaper) by studying the characteristic properties of texture—self similarity and scale variability. Finally, we describe two methods for recovering the 3D geometry and reflectance properties of an object from images taken under natural illumination. We note that the structure of the surrounding environment, modulated by the reflectance, becomes embedded in the appearance of the object giving strong clues about the object's shape.

Though these domains are quite diverse, an essential premise—that observations of objects contain within them salient clues about the object's structure—enables new and powerful approaches. For each problem we begin by investigating what these clues are. We then derive models and methods to canonically represent these clues and enable their full exploitation. The wide-ranging success of each method shows the importance of our carefully formulated observations about geometry, and the fundamental role geometry plays in visual perception.

Chapter 1: Introduction

1.1 Background and motivation

The beauty of our world is due in part to the staggering diversity of appearance that objects and places may have. Modeling the vast complexity of this phenomenon to empower machines with vision is consequently a daunting task. Though many factors contribute to appearance, we observe that geometry plays a uniquely fundamental role.

Geometry reconstruction is at the heart of many important problems. Pragmatically understanding how the world around us is physically shaped is a necessary first step in enabling robotic interaction; gives important clues to scene analysis, object detection, gesture recognition and semantic segmentation; and enriches our understanding of human visual perception. It is no surprise then, that this problem has attracted wide attention.

Due to the complex nature of geometry reconstruction problems, past work has focused on casting the problems into simplistic models. In this thesis we focus, instead, on how the complexity itself can be used to our advantage. By noting that important clues about underlying geometry become embedded in appearance, we derive solutions to previously unapproached problems, or introduce new ways to look at longstanding problems.

> *How can we recover the geometric deformation of an object by looking only at its appearance before and after the deformation?*

Image registration—the process of aligning one image to another—has important applications to many problems including image stabilization [1], subject tracking [2,3], and medical imaging [4], to name a few. Past work has focused primarily on the visual similarity of the registered image and the target, under-emphasizing the importance of recovering a physically plausible deformation field. We seek to recover a deformation that conforms to the underlying geometry of the subject. By doing so, we are able to model large deformations, such as a person smiling.

How can we recover the geometry of an object that is broken?

Archaeologists and forensic scientists spend vast amounts of time in the tedious process of reassembling broken objects. In order for the process of historical or criminal analysis to proceed, we must first reconstruct the artifacts. In many cases, the fragments themselves are locked away in warehouses, inaccessible except by special request. By performing reassembly virtually, we greatly accelerate this process, increase access, and ensure that once reassembled, the data is as accessible as possible, without any risk to further damaging the object or any subtle details. Past work in this field has simplified the problem by assuming that the object belongs to a specific class (vases, or bowls) or that it is richly painted. We avoid any such limiting assumption by focusing only on the boundary of each fragment.

How can we recover and represent the geometry of a texture?

Texture analysis has important applications in scene segmentation, industrial surface inspection, and remote sensing. For example, empowering a robot with the ability to perceive texture can enable it to drive around a patch of dangerously loose sand. The bulk of past work has approached this important problem by inspecting the *appearance* of texture, which can change dramatically. A gravel road, for example, may appear completely differently if it is wet, covered in a dusting of snow, or reflecting the sun's glare. Our approach is instead to focus on the underlying *geometry* of texture, which does not change.

> *How can we recover the geometry of an object with non-trivial reflectance properties as it appears in the wild?*

Today, for the first time in history, consumers can print replications of objects with in-home 3D printers. Unfortunately, creating the 3D model itself is something that cannot be done at home, forcing users to download models that may not be exactly what they need, or to manually create models using advanced software packages. Enabling users to use the camera that they already own to create 3D models is the missing link that will bring the vast possibilities of 3D printing into the mainstream. We live in a world made of diverse and interesting materials that result in complex appearance. Though past work has avoided this complexity by focusing on controlled lighting or a small set of materials, we focus instead on fully exploiting the complexity. By bringing 3D geometry recovery into the wild, we enable affordable and safe 3D acquisition of objects without requiring that we bring them into a laboratory, or paint them to make the problem easier. This has profound implications in cultural heritage preservation, 3D printing, robotic navigation, and many other fields.

1.2 Related work

Each of the problems we approach has received much related attention. In the following chapters we include more detail about the related work. Here we will briefly summarize.

1.2.1 Geometric deformation recovery

Image registration methods are split into two categories. Feature based approaches utilize a sparse set of control points to drive the registration process [5–7]. These control points, which are typically manually defined, serve as known point correspondences. Once these points are aligned with, the rest of the deformation field is then interpolated. Because of the interpolation, accuracy necessarily decreases as we move away from the control points.

The second high-level approach is to utilize the full image in registration [8–10]. These methods are called "intensity based" since they rely on the difference in intensities as the driving factor of the registration, and do not require any manual intervention. Since these methods do not rely on any known correspondences, however, there is no guarantee that physically meaningful points will become aligned with each other.

What is missing from both approaches is an appreciation for the underlying *geometry* of the imaged subject. By treating the intensity structure as a proxy for the underlying structure of the object, we can ensure that salient geometric features remain intact as the deformation proceeds. The result is a dense set of correspondences that automatically results in high-accuracy alignment of key feature points.

In Section 2.2 we give a more comprehensive review of the literature.

1.2.2 Fragmented geometry reassembly

Past work in fragmented object reassembly has focused on volumetric objects [11,12]. When such objects are broken, richly textured "break surfaces" are revealed that have matching geometry with the fragments they adjoin to. By analyzing these surfaces we can find and align matching fragments.

For thin objects, or in cases where it is easier to simply image the exterior surface of the fragment, we do not have such break surfaces. The problems of finding matching fragments, and aligning them both become much more ambiguous. To deal with the ambiguity, a common approach is to assume that the object is known to belong to a certain class, most commonly rotationally symmetric objects (such as vases and bowls) [13–15]. Others make assumptions about the painted texture of the object [16].

We avoid making any such assumptions by focusing only on the boundary of each fragment. Since matching fragments must share similar geometry and photometry along this thin strip, we make full use of it by deriving a novel representation of its scale space that enables rapid detection of matching fragments.

Section 3.2 reviews the literature in more detail.

1.2.3 Geometric texture analysis

Past work in texture classification and representation has focused almost exclusively on its appearance [17, 18]. This notion, though intuitive, is difficult to generalize because appearance depends on external factors such as lighting and viewing conditions. Though some work has tried to address this variability [19, 20], no past work has approached the problem directly from a geometric perspective as we do.

By observing that texture is defined by its repetitious geometric structure, we derive a novel method for estimating the geometry that underlies its appearance. We then introduce a method for compactly representing the scale-variability of the geometry in a rotation-invariant way so that we may perform classification.

Section 4.2 provides a detailed review of the related literature.

1.2.4 Geometry recovery in the wild

Geometry recovery is one of the most longstanding problems of computer vision, with decades of past work. Though our world is full of objects made with many materials with complex reflectance properties situated in beautiful illumination environments, past work has framed the problem into simplistic models of light and reflectance.

Traditionally, geometry recovery has been constrained to the dark room. Past work has found that by controlling light, many nice properties arise in appearance that can be used to simplify the problem [21–23].

When we step outside of the dark room, we replace the single point light with a full sphere of incoming light. In the face of this complexity, past work has focused on a theoretical material property—Lambertian reflectance. Tough theoretical, materials such as stone and rough paper have been shown to exhibit properties that enable shape estimation. Recent work on single image geometry recovery [24–26] has shown that for real-world illumination, the appearance of such materials takes a parametric form. In the case where multiple images are available, work in structure from motion uses the property of *viewpoint independent appearance* to triangulate sparse 3D scene features [27].

Our work is unique in that we allow for non-trivial reflectance properties, and complex natural illumination. Though the problem we approach is challenging, we show how the surrounding illumination environment, modulated by the reflectance becomes encoded as latent structures in the observation images. By extracting and exploiting these structures we are able to estimate both the reflectance and shape of the object.

Section 5.2 gives a more in depth review of the related work for both the single image case, and the case in which multiple images are available.

1.3 Contributions

Taken as a whole, this dissertation demonstrates how fundamental geometric properties become embedded as latent structures in observations, and how these structures can then be extracted and used to help tackle important problems in visual perception. Several concrete theoretical and practical contributions are presented.

- **An intensity-based image registration formulation designed to leverage geometric intensity structure.**

  We observe that the intensity structure of many types of images directly encodes the structure of the imaged surface. By treating the image as a 2D membrane in a 3D space we then derive a novel, physically-based approach to registration.

- **Curvature-weighted likelihood formulation for image registration.**

  At each pixel of the template image we construct a Gaussian distribution with a variance proportional to the membrane's original curvature. This enables image structures to drive the registration process.

- **Membrane geometry preserving constraints.**

  We model the inherent flexibility and elasticity of the image membrane by penalizing changes in surface bending and stretching potential energy. By weighting the constraints based on the membrane's initial curvature we ensure that image structures keep their shape as the surface deforms.

- **An image-registration metric that evaluates how key features are aligned at a sub-pixel level.**

  By manually annotating features and then performing registration on low-resolution images, we can track not just how similar the resulting image is to the target, but also how accurately the key features align with their targets. This metric evaluates how well the actual structure of the underlying object is maintained, penalizing deformations that distort the subject for the sole purpose of a more visually pleasing result.

- **A framework for reassembling fragmented thin-shelled objects with minimal user interaction.**

  Our overall framework for reassembling objects using only the boundary of each fragment utilizes our novel fragment representation formulation to determine candidate matches. It then filters these matches by investigating how well the fragments align.

Utilizing user feedback to cull the otherwise exponential search space, it proceeds only when a positive match is found. As the vessel is reconstructed the additional context given by the relationship between the matched fragments is used to inform future matches. In brief, the user and the machine are both utilized for their strengths.

- **A scale-space representation of the geometry and photometry of the boundary contours of object fragments.**

  By encoding the scale variability of the boundary contours shape and color as a multichannel image, we compactly represent the full scale-space of the fragment boundaries. This enables rapid estimation of matches and verification of matches.

- **An image-registration based method for finding matching fragments.**

  Utilizing the coarse scale to quickly find candidate matches, and the finer scale detail to refine and validate the matches, our image-registration based method rapidly estimates fragments that are likely to align.

- **A framework for incremental global optimization of fragment alignments in complex reassemblies.**

  The relationship between fragments as they are found and added to the reassembly form a graph. By detecting cycles in the graph we are able to determine sets of fragments whose alignment accuracies are interdependent. By jointly re-optimizing their alignments we incrementally find a globally optimal result, without having to perform a full global optimization from scratch.

- **A photometric stereo method uniquely tailored for geometric texture.**

  By exploiting the characteristic qualities of texture—scale variability and self similarity—we derive a novel photometric stereo method design specifically for textures. We detect geometrically disparate neighborhoods of similarly oriented pixels by inspecting intensity profiles. Then, when estimating the orientation of each of these pixels, the information about each can help inform the orientation of the others.

- **A rotation-invariant scale-space representation of geometric texture.**

  Using a canonical encoding pipeline, we derive a compact representation of the scale-space of geometric texture. The representation is invariant to in-plane rotation, and robust to a degree of out-of-plane rotation. We evaluate the representation by preforming hundreds of classification experiments.

- **An extensive texture database consisting of more than $40,000$ images under varying illumination and rotation angles.**

  In order to evaluate our own work, and to stimulate future work on geometric texture, we acquired a massive database of 20 textures imaged at thee distances with many combinations of both in-plane and out-of plane rotations. In each setting, each texture is imaged approximately 65 times with different lighting directions which themselves are recorded.

- **A probabilistic framework for jointly estimating the shape and reflectance of an object from a singe image with known, but unconstrained natural illumination.**

  Our work is the first to approach the problem of estimating the shape of an object with a non-trivial reflectance situation in complex natural illumination. We show how complex appearance contains within it clues about the geometry and reflectance properties. We formulate the problem probabilistically and derive canonical constraints to account for ambiguous appearance, and ensure a realistic reflectance estimate.

- **A multi-scale optimization framework for refining single-view shape estimation.**

  Due to the reliance on the necessarily non-parametric illumination, the optimization problem itself takes on a non-parametric form. In order to find a globally optimal solution, we introduce a multi-scale framework, and use a discrete optimization formulation to find the solution at each scale.

- **A probabilistic framework for jointly estimating full 3D shape and reflectance from multiple images.**

  Moving beyond a single image, our work is also the first to focus on full 3D geometry recovery for object with non-trivial reflectance in natural illumination. We utilize a triangle mesh, initialized from the visual hull of the observation silhouettes, to link the observations. We then show how the information from each observation can be combined to tighten the distribution of possible orientations that each facet may have. A novel constraint on the local curvature is introduced to account for regions that are not visible in any observations, or take on an ambiguous appearance in all images.

- **Two databases of calibrated images of real-world objects taken in many scenes and the associated ground-truth geometry.**

  To evaluate our geometry recovery algorithms, and to inspire future work we have acquired two new data-sets. The first includes single images of six objects in four different illumination environments. The second consists of multiple (approximately 12) observations of four different objects in three illumination environments. In both cases ground truth geometry was acquired using a light-stripe range scanner, and aligned to the data to enable a rigorous evaluation.

1.4 Overview

In this thesis we approach several problems of geometry reconstruction. Each problem consists of a set of observations, such as images or 3D data, and asks a question about the underlying geometry. The remainder of the document proceeds as follows.

Geometric deformation recovery

In Chapter 2 we approach the problem of appearance variation due to a geometric deformation. A smile, for example, can dramatically change how a person appears. We observe that despite this change in appearance, the salient geometric features (e.g. nose, mouth, eyes) remain much the same. In many cases these geometric structures become encoded as intensity structures in the image. This enables us to recast the problem of extracting the underlying deformation as a problem of 3D surface alignment. Using constraints derived to keep the structures intact, we are able to accurately register images that have undergone significant deformations, while ensuring that the recovered motion itself is physically plausible. We derive these constraints as curvature weighted first and second order derivatives that correspond to the changes in stretching and bending potential energies of the 3D surface. Experimental results on real data demonstrate the effectiveness of our method, in particular, its robustness to local minima and its ability to establish accurate correspondences across the entire image.

Fragmented geometry recovery

In Chapter 3 we approach the problem of fragmented geometry recovery. When an object, such as a vase, statue, or bone, becomes broken all that remains is partial, fragmentary views of its original geometry. We show that, contrary to past work, it is not necessary to make restrictive assumptions about the shape or painted texture of the object. Instead, we rely only on the certainty that adjoining fragments will share similar geometric and photometric structure embedded along and across their matching boundaries. We begin by encoding the scale variability of each fragment's boundary contour in a multi-channel, 2D image representation. Using this multi-channel boundary contour representation, we identify matching sub-contours via 2D partial image registration. We then align the fragments by minimizing the distance between their adjoining regions while simultaneously ensuring geometric continuity across them. The configuration of the fragments as they are incrementally matched and aligned form a graph structure that we use to improve subsequent matches. By detecting cycles in this graph, we identify subsets of fragments with interdependent alignments. We then minimize the error within the subsets to achieve a globally optimal alignment. We leverage user feedback to cull the otherwise exponential search space; after each new match is found and aligned, it is presented to a user for confirmation or rejection. Using ceramic pottery as the driving example, we demonstrate the accuracy and efficiency of our method on six real-world datasets.

Geometric texture analysis

In Chapter 4 we bring our study of intrinsic structure to the domain of geometric textures (i.e. gravel, bark, carpet, sandpaper, etc.). Past work has focused on the appearance of texture which will change as the viewing and illumination conditions change. Instead we focus on the underlying geometry, which remains constant. Textures, whether they are regular or stochastic, exhibit some form of repetition in their underlying geometry. We show how the fundamental properties of texture geometry—self similarity, and scale variability—become embedded in images of their surfaces as latent structures. Using basic observations about the scale variability of texture geometry, we derive a compact, rotation invariant, scale-space representation of geometric texture. To evaluate this representation we introduce an extensive new texture database that contains multiple distances as well as in-plane and out-of plane rotations. The high accuracy of the classification results indicate the descriptive yet compact nature of our texture representation, and demonstrates the importance of geometric texture analysis, pointing the way towards improvements in appearance modeling and synthesis.

Geometry recovery in the wild

Finally, in Chapter 5 we discuss two methods for recovering the 3D geometry and reflectance properties of an object situated in natural illumination, one in cases where only a single image is available, and another for when there are multiple images. We exploit the structure of the surrounding illumination environment that, modulated by the reflectance, has become embedded in the appearance of the object. Sparse but salient orientation clues (such as the reflected sun) give a strong indication about local geometry, while less unique regions (such as the sky, or grass) give only weak clues. In the single image case, such ambiguity is resolved though carefully designed constraints. In the multiple image case, the ambiguity is reduced when multiple images corroborate a tighter range of possible orientations. Through extensive evaluation on synthetic and real-world data, we demonstrate that for the wide range of real-world reflectance and illumination environments, our methods are able to accurately recover geometry.

In Chapter 6 we discuss our conclusions and opportunities for future work.

Chapter 2: Geometric deformation recovery

2.1 Introduction

When we look at two images, say of a person staring blankly, and then smiling widely, we immediately know that it is the same person. Further, we intuitively know which parts of the image have stretched, which have compressed, which parts have moved, and which have stayed in the same position. This is even true for less familiar objects. Without any knowledge of how a brain functions, we can observe MRI scans and, by comparing the appearance structure, we can quickly begin to label which features correspond to each other, and how the tissue may be distorted to explain the change in appearance.

But for a computer an image is just a set of pixels. There is no inherent notion of "nose" or "eye" embedded in the data. Of course, two images can simply be subtracted from each other and the raw intensity difference can be compared, but non-rigid deformations such as smiling may then yield a higher overall difference values than images of different people entirely. In order to compare the actual imaged object then, we must create a distance measure that is invariant to complex, local distortions.

Image registration is the process of aligning a template image to a reference image by locally deforming the template image. In addition to enabling a more meaningful comparison of the object in the two images, modeling nonlinear, local deformations has important applications in many computer vision problems including image stabilization [1], subject tracking [2,3], and medical imaging [4], to name a few.

There are two primary approaches to nonrigid image registration: intensity-based and feature-based. Intensity-based approaches [8–10] attempt to minimize the intensity differences across the entire image. Such methods produce dense correspondences but suffer from ambiguities arising from similar intensity regions. Feature-based methods [5–7] compute deformations that align a sparse set of specifically selected features. These points are then used in conjunction with a parametric model to interpolate the recovered deformations across the rest of the image. In addition to the separate challenge of detecting and matching good features (which often relies on manual intervention), the overall quality of the registration directly relies on the interpolation method. Consequently, accuracy inherently decays rapidly as the distance from the feature points increases.

In this paper, we introduce an automatic nonrigid 2D image registration method that establishes dense and accurate correspondences across the entire image without the need to provide feature correspondences a priori. Our key idea is to model the image as a 2D membrane embedded in a 3D spatial-intensity space. We then formulate nonrigid image registration as the process of aligning two membranes by deforming one to the other while preserving its local geometric structures. In particular, we model the elastic and bending potential energies of the membrane. By penalizing their changes, the local structures of the template membrane are preserved as it deforms to meet the reference membrane.

We derive a probabilistic formulation of this membrane nonrigid image registration. We model each template image point as a Gaussian and seek the maximum a posteriori estimate of the template image as a mixture of Gaussians given the reference image. Our main contributions are a newly derived likelihood and priors that reflect physically-motivated constraints on the membrane geometry:

Novel likelihood: We construct a Gaussian at each pixel of the template image scaled by the membrane's original curvature at that point. This naturally encodes the significance of the underlying image structure, which in turn encourages features to align with corresponding features.

Bending energy: We model the inherent flexibility of a membrane by penalizing local surface deformations in proportion to the membrane's original curvature. This corresponds to minimizing the change in potential bending energy which translates into a novel curvature-weighted second order derivative prior.

Stretching energy: We model the inherent elasticity of a membrane by penalizing surface stretching and compression. This corresponds to minimizing the change in potential elastic energy across the membrane which translates into a novel first order derivative prior.

Intuitively, this formulation leads to surface regions with prominent local structures (features of the membrane) to be preserved and aligned with each other while more smooth regions are allowed to deform more flexibly. By preserving the shape of the membrane features, their appearance in the image being modeled remain true to their underlying geometry.

We demonstrate the accuracy and effectiveness of our method on 2D slices of real brain MRIs and images of faces with different expressions. In particular, we show that in addition to a significant decrease in overall intensity error, our method establishes accurate correspondences of prominent image structures automatically. This has strong implications in various applications since local image structures usually correspond to meaningful geometric structures of the imaged scene or object, and accurately aligning such structures is of great importance.

2.2 Related work

Nonrigid image registration has been a popular area of research. Here we focus on methods that specifically address the shortcomings of both intensity-based and feature-based methods. We refer the reader to surveys of the rich literature [4, 28, 29] for more thorough context.

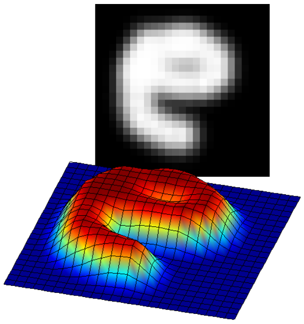
Fischer and Modersitzki [30] combine the two approaches on a sliding scale. They initially register a set of manually established features, then incrementally shift towards a uniform intensity-based metric. Our curvature-scaled objective function has a similar effect in that it encourages the rapid registration of feature rich areas. It does so, however, without requiring predefined features or by giving priority to the registration of any subregion.

Fischer and Modersitzki [31] also introduced a "curvature-based" normalization term that encourages locally smooth deformations by penalizing sharp changes in the displacement field. Although we also describe our bending energy constraint term as "curvature-based," the two approaches are fundamentally different. Whereas their normalization term is a second-order derivative of the 2D displacement field, we impose an energy minimization prior on the membrane, i.e., the image modeled as a 3D spatial-intensity surface. This added dimension allows us to impose geometrically-induced constraints on the image deformation.

Intensity-based methods assume that corresponding regions in the imaged scene maintain the same intensity pattern in both images. Previous authors [32, 33] have noted that this assumption can lead to violations of the basic physical properties of the subject which are present despite changes in illumination. To address this they use mass or volumetric constraints specific to their given applications. More general methods like Thirion's Demons method [8] and the recent diffeomorphic extension of this work by Vercauteren et al. [10] smooth the 2D deformation field thereby preventing large feature displacements from tearing or folding the deformation field. Although smooth deformation fields are found, ambiguities arising from similar intensity patterns of non-corresponding regions result in undesirable non-local artifacts. In our physically motivated model, we avoid such local minima by preserving the shape of the image membrane thereby maintaining local structures as they move across the image. To address folding we introduce a novel prior which allows pixels to come quite close to each other without overlapping. This allows us to model the common physical occurrence of creasing which is impossible under the various smoothing models.

Recently, probabilistic formulations of nonrigid image registration have gained further attention. Jian and Vemuri [34] use a Gaussian mixture model to register two point sets by placing a Gaussian at each point. Our work is most closely related to the extension of this approach by Myronenko et al. [9] that formulates image registration as a Gaussian mixture estimation with Gaussians centered on each pixel. By placing quadratic priors on each Gaussian, they preserve the distance of each pixel to its neighbors thereby avoiding tearing and folding of the deformation field. This results in a locally smooth deformation with well-minimized intensity distance on synthetic deformations. Unavoidably, however, these priors perform less well on real-world data which exhibit more complex transformations that cannot be modeled with assumptions of smoothness. Since accurate correspondences are of primary concern in many applications, we show that the minimization of an intensity distance is an insufficient objective function without shape preserving constraints.

2.3 Bayesian membrane registration



We model the image as a 2D membrane in a 3D space. In order to ensure this membrane approximates the actual imaged surface, the intensities are normalized and the height of each pixel is set proportionately to the log of the normalized intensity.[1] As noted by Koenderink and van Doorn [35], by using the log-intensity we

[1] Results are consistent so long as the images are normalized consistently. Since this formulation is geometrically invariant, the scale only effects the convergence rate.

ensure a geometrically invariant intensity encoding which eliminates any effect intensity magnitude may otherwise have on our geometric constraints while simultaneously achieving a degree of symmetry between the Cartesian pixel coordinates and heights of the points on the membrane.

More precisely, we view the image coordinates $\widehat{\boldsymbol{x}} = (\widehat{\boldsymbol{x}}_u, \widehat{\boldsymbol{x}}_v)$ and scaled logarithm of the normalized intensity of each point $I(\widehat{\boldsymbol{x}})$ together as points $\boldsymbol{x} = (\widehat{\boldsymbol{x}}_u, \widehat{\boldsymbol{x}}_v, I(\widehat{\boldsymbol{x}}))$ of a 2D membrane in a 3D space. In many cases we may assume that this membrane reflects the geometry of the imaged object. For instance, a Lambertian surface would have its normals roughly encoded in its shading and the intensity in medical images reflects the density of the subject.

Similar to Myronenko et al. [9], we formulate nonrigid image registration as a MAP estimation of a product of Gaussian mixture densities. The posterior, representing the probability of the template image $\boldsymbol{Y}$ given the reference image $\boldsymbol{X}$ and parameters θ, is formulated as

$$p(\boldsymbol{Y}|\boldsymbol{X}, \theta) \propto p(\boldsymbol{X}|\boldsymbol{Y}, \theta)p(\boldsymbol{Y}|\theta) \ , \tag{2.1}$$

where we assume uniform normalization $p(\boldsymbol{X})$. We have five parameters, $\theta = (\boldsymbol{Y}^0, \sigma_0, \beta_\mathrm{e}, \beta_\mathrm{b}, \beta_\mathrm{f})$, which we describe below. Here $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^\mathrm{T}$ is an $N \times 3$ matrix containing the points in the reference membrane $\boldsymbol{x} = (\widehat{\boldsymbol{x}}_u, \widehat{\boldsymbol{x}}_v, I(\widehat{\boldsymbol{x}}))$ and $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M)^\mathrm{T}$ is an $M \times 3$ matrix containing the final locations of the registered template membrane's points $\boldsymbol{y} = (\widehat{\boldsymbol{y}}_u, \widehat{\boldsymbol{y}}_v, I(\widehat{\boldsymbol{y}}))$. We denote the original, undeformed template membrane as $\boldsymbol{Y}^0 = (\boldsymbol{y}^0_1, \ldots, \boldsymbol{y}^0_M)^\mathrm{T}$. N and M are the number of pixels in the images (which need not be equal in size). We model the likelihood as a product of N independent Gaussian mixture densities $p(\boldsymbol{X}|\boldsymbol{Y}, \theta) =$

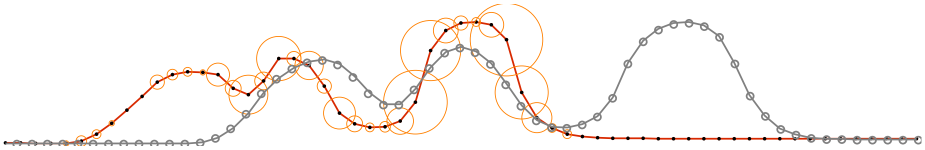


Figure 2.1: 1D initialization Shown in this 1D example, a Gaussian is established for each pixel of the template image (solid) with standard deviation (circles) proportional to the curvature at that point. This allows prominent local structures that usually have high curvature to travel further and align with corresponding structures of the reference image (dotted) while preserving their shapes as modeled by their elastic and bending energies.

$\prod_n p(\boldsymbol{x}_n|\boldsymbol{Y})$. Building on this formulation, we introduce a curvature-based scaling to each point as we discuss next.

Our key contributions lie in the three priors on the template membrane $\boldsymbol{Y}$,

$$p(\boldsymbol{Y}|\theta) \propto \exp\left(-\beta_{\mathrm{e}}\mathcal{E}(\boldsymbol{Y}) - \beta_{\mathrm{b}}\mathcal{B}(\boldsymbol{Y}) - \beta_{\mathrm{f}}\mathcal{F}(\boldsymbol{Y})\right) \ . \tag{2.2}$$

The first, $\mathcal{E}(\cdot)$, quantifies the amount of change in elastic potential energy in the membrane. The second, $\mathcal{B}(\cdot)$, quantifies the change in bending potential energy in the membrane. Finally, $\mathcal{F}(\cdot)$ quantifies the amount of folding, or overlap, in the membrane. Each function is weighted by a parameter $\beta_{\{\mathrm{e,b,f}\}} \in \theta$. We will now describe each component of the posterior in more detail.

2.3.1 Gaussian Mixture Likelihood

The mixture density $p(\boldsymbol{x}_n)$ for a pixel of the reference image $\boldsymbol{x}_n$ is expressed probabilistically as a Gaussian mixture where each point of the template membrane is expressed as its own Gaussian distribution

$$p(\boldsymbol{x}_n|\boldsymbol{Y}) = \sum_{m=1}^{M} \frac{1}{M}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) \ . \tag{2.3}$$

Observing that regions with prominent local structures (features) are more indicative of the membrane's overall shape, we allow points in these regions a larger range of motion by scaling the Gaussian centered at each point by the membrane's original curvature at that point. Using the squared mean curvature $H^2(\boldsymbol{y}_m^0)$ we model this with a per-point mean and standard deviation of

$$\boldsymbol{\mu}_m = \boldsymbol{y}_m, \qquad \boldsymbol{\Sigma}_m = (H^2(\boldsymbol{y}_m^0)\sigma_0)^2 \boldsymbol{I}_3 \ , \tag{2.4}$$

where $\boldsymbol{I}_3$ is the identity matrix as each image dimension is statistically independent. These feature rich areas maintain their shape due to increased rigidity constraints (discussed in the next section). Intuitively, this leads to feature-rich surface regions to be preserved and aligned with each other, guiding the registration of the rest of the membrane.

We can then express the likelihood across the entire image as an unweighted product of these Gaussian mixture densities

$$p(\boldsymbol{X}|\boldsymbol{Y}, \sigma_0) \propto \prod_{n=1}^{N} \sum_{m=1}^{M} \exp\left[-\frac{1}{2} \left\| \frac{\boldsymbol{x}_n - \boldsymbol{y}_m}{H^2(\boldsymbol{y}_m^0)\sigma_0} \right\|^2 \right] \ . \tag{2.5}$$

In other words, for a given scale parameter σ_0, the final pixel locations $\boldsymbol{Y}$ that maximize this likelihood represent the deformation that maps the points of the template membrane to regions of the reference membrane.

In Figure 2.1 we show a simple one-dimensional, $(x, I(x))$, example where the initial template $\boldsymbol{Y}^0$ is shown in red, and the reference $\boldsymbol{X}$ is shown in gray. The relative standard deviations of the Gaussians are shown as orange circles. As shown in Figure 2.2, this increase in the search space for key regions of the curve is necessary to avoid local minima and preserve the geometry of membrane features.

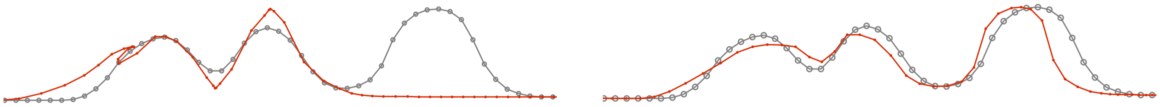


Figure 2.2: 1D registration example Two results after registering the curves of Figure 2.1 are shown (solid) relative to the target curve (dotted). Standard smoothing priors [9] (*left*) can cause local minima to be found. Here an entire peak is unregistered while two peaks have collapsed into one. Imposing our physically-based constraints (*right*) ensures that the structure of the entire curve is maintained during deformation resulting in a more accurate registration.

The shape of the deformed membrane must now be considered. Without constraints on the deformation, the pixel locations can be permuted at will to maximize the likelihood. To ensure an accurate deformation, we introduce physically-motivated priors that operate on the local geometry of the membrane.

2.3.2 Shape-Preserving Priors

The membrane model of an image allows us to incorporate physically-based constraints that preserve the local intensity structures of the image as it deforms. In particular, we model the elastic and bending potential energies of the membrane and impose geometric constraints that minimize the changes in these energies.

Elastic Energy The elastic energy of a deformation captures the change in elastic potential as the membrane deforms. We define this energy as the sum of the elastic energy across all points $\mathcal{E}(\boldsymbol{Y}) = \sum_m \mathcal{E}(\boldsymbol{y}_m)$. We define the elastic energy at a point as the change in elastic potential energy at that point $\boldsymbol{y}$ relative to the potential at that point in the original membrane $\boldsymbol{y}^0$. We evaluate the potential of a point on a membrane using Hooke's law $E = \frac{1}{2}kx^2$. By assuming the elastic constant (k in Hooke's law) is uniform across the membrane we let $\beta_\mathrm{e} = (k/2)^2$ which is then used to weight the entire energy term. The relative displacement (x in Hooke's law) at each point naturally corresponds to the total change in distance to the point's neighbors $\mathrm{ne}(\boldsymbol{y})$.

By squaring the difference in potential of the relaxed and deformed membranes, we naturally quantify the amount of elastic energy at each point as

$$\mathcal{E}(\boldsymbol{y}) = \sum_{\boldsymbol{y}_i \in \mathrm{ne}(\boldsymbol{y})} (\left\|\boldsymbol{y}_i - \boldsymbol{y}\right\|^2 - \left\|\boldsymbol{y}_i^0 - \boldsymbol{y}^0\right\|^2)^2. \tag{2.6}$$

Note that because the intensity of a pixel does not change this reduces to

$$\mathcal{E}(\boldsymbol{y}) = \sum_{\boldsymbol{y}_i \in \mathrm{ne}(\boldsymbol{y})} (\|\widehat{\boldsymbol{y}}_i - \widehat{\boldsymbol{y}}\|^2 - 1)^2. \tag{2.7}$$

This prior differs considerably from stretching or elastic constraints of past work. Specifically, the first-order smoothing terms used in past work impose smoothing on the 2D deformation field itself, necessarily resulting in overly smooth local deformations.

Bending Energy We also model the bending potential energy of the membrane and derive an energy term which quantifies the change in this potential as the membrane deforms. We define the total bending energy as the sum across all points, $\mathcal{B}(\boldsymbol{Y}) = \sum_m \mathcal{B}(\boldsymbol{y}_m)$. Our bending potential function is based on the Willmore energy $\int_S \frac{1}{2}H^2 - KdA$, where H is the mean curvature function and K is the Gaussian curvature function. By the Gauss-Bonnet theorem K is a topological invariant, and so remains constant during the deformation. Since we are concerned with the change in this energy, this term cancels out. We extend the Willmore energy to include the inherent rigidity of structural features by considering the potential of each point separately.

Whereas homogeneous membranes have uniform elasticity, the flexibility of a membrane varies with the curvature of the undeformed surface [36]. This translates to weighting the bending energy with a per-point rigidity coefficient equal to the squared mean curvature of the undeformed membrane at that point $H^2(\boldsymbol{Y}^0)$. This term also provides robustness to noise since a corrupt pixel will yield a high curvature value at that point. Since mean curvature is computationally expensive, we use the Laplacian $\Delta(\cdot)$ as an approximation for $H^2(\cdot)$ when computing the change in energy [37]. We define the bending energy as the weighted squared change in bending potential

$$\mathcal{B}(\boldsymbol{y}) = H^2(\boldsymbol{y}^0)\left(\Delta(\boldsymbol{y}) - \Delta(\boldsymbol{y}^0)\right)^2 . \tag{2.8}$$

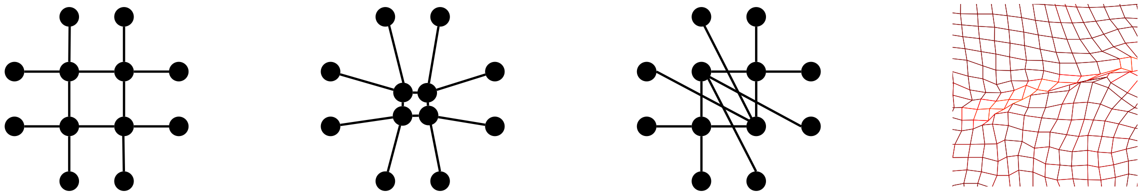
At a given point, the Laplacian of a surface is expressed using the (log) intensity heights $I(\cdot)$ of the point $\boldsymbol{y} = (\widehat{\boldsymbol{y}}_u, \widehat{\boldsymbol{y}}_v, I(\widehat{\boldsymbol{y}}))$ and its negative direction and positive direction neighbors $\boldsymbol{y}_-$ and $\boldsymbol{y}_+$ respectively

$$\Delta(\boldsymbol{y}) = \left(\frac{h_-(I(\widehat{\boldsymbol{y}}_+) - I(\widehat{\boldsymbol{y}})) - h_+(I(\widehat{\boldsymbol{y}}) - I(\widehat{\boldsymbol{y}}_-))}{h_+ h_- h_\pm}\right)^2 , \tag{2.9}$$

where the distance to the positive direction neighbor h_+ the negative direction neighbor h_- and the distance between midpoints $h_\pm$ are used

$$h_+ = \|\widehat{\boldsymbol{y}}_+ - \widehat{\boldsymbol{y}}\| \ , \quad h_- = \|\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{y}}_-\| \ , \quad h_\pm = \|\left[(\widehat{\boldsymbol{y}}_+ + \widehat{\boldsymbol{y}}) - (\widehat{\boldsymbol{y}} + \widehat{\boldsymbol{y}}_-)\right]/2\| \ . \tag{2.10}$$

As a time saving approximation we assume $h_+ = h_- = h_\pm$. We also note that the numerator is equal for $\Delta(\boldsymbol{y})$ and $\Delta(\boldsymbol{y}^0)$ since the intensities of the pixels do not change. Further, we note that $h^0_\pm$ is constant which allows us to reduce the horizontal bending



(a) $\mathcal{E}\ =\ 0.0$ **(b)** $\mathcal{E}\ =\ 7.7$ **(c)** $\mathcal{E}\ =\ 5.7$ **(d)** Crease

Figure 2.3: Avoiding folding The elastic penalization for areas of compression is minimized when neighboring surface patches fold over one another in featureless regions. We address this with an explicit prior on folding which allows for creases to form but eliminates folding.

penalization of Equation 2.8 to

$$\mathcal{B}(\boldsymbol{y}) \propto H^2(\boldsymbol{y}^0)\left(h_{\pm}^{-2}-1\right)^2 . \tag{2.11}$$

For 2D images we consider horizontal, vertical, and two diagonal bending energies by formulating $\mathcal{B}_{\rightarrow}$, $\mathcal{B}_{\downarrow}$, $\mathcal{B}_{\nearrow}$, and $\mathcal{B}_{\searrow}$ analogously and take the sum

$$\mathcal{B}(\boldsymbol{y}) = \mathcal{B}_{\rightarrow}(\boldsymbol{y}) + \mathcal{B}_{\downarrow}(\boldsymbol{y}) + \mathcal{B}_{\nearrow}(\boldsymbol{y}) + \mathcal{B}_{\searrow}(\boldsymbol{y}) \ . \tag{2.12}$$

Folding Prior During registration, regions of the deforming template membrane will expand and compress to meet the corresponding reference regions. As shown in Figure 2.3, since our elastic energy constraint encourages uniform spacing and our bending energy constraint applies primarily to feature rich areas, folding can occur. Although the bending prior discourages this in textured areas, it is not sufficient in relatively featureless regions.

Conventional methods decrease this by imposing second order derivative penalizations on the 2D deformation field [9,31] or by specifically modeling diffeomorphic registrations [10]. Problems arise, however, in regions that change in size dramatically. As real-world objects inevitably experience such large deformations, a more accurate model should allow sharp boundaries in the deformation field as neighboring regions converge and creases form.

We allow such sharp boundaries to form with an explicit model of folding that allows pixels to come quite close to each other without penalty while strongly penalizing folding. We model this with a sigmoid function on each of the four neighboring directions of a point $\boldsymbol{y} = (\widehat{\boldsymbol{y}}_u, \widehat{\boldsymbol{y}}_v, I(\widehat{\boldsymbol{y}}))$. The folding energy of a deformation is then the sum across the deformation of each of these four values

$$\mathcal{F}(\boldsymbol{Y}) = \sum_{m=1}^{M} \left(\mathcal{F}_{\rightarrow}(\boldsymbol{y}_m) + \mathcal{F}_{\leftarrow}(\boldsymbol{y}_m) + \mathcal{F}_{\uparrow}(\boldsymbol{y}_m) + \mathcal{F}_{\downarrow}(\boldsymbol{y}_m)\right) \ . \tag{2.13}$$

For example, the right neighbor function is given by

$$\mathcal{F}_{\rightarrow}(\boldsymbol{y}) = \left(1 + \exp\{c(\widehat{\boldsymbol{y}}_u^{+} - \widehat{\boldsymbol{y}}_u + t)\}\right)^{-1}, \tag{2.14}$$

where $\widehat{\boldsymbol{y}}^{+}$ is the right neighbor of $\widehat{\boldsymbol{y}}$. We establish the other three functions similarly. In this formulation a sufficiently high value for c and low value for t effectively make this a step function that penalizes the folding of neighboring pixels while allowing pixels to form sharp boundaries without penalty.

2.3.3 MAP Estimation

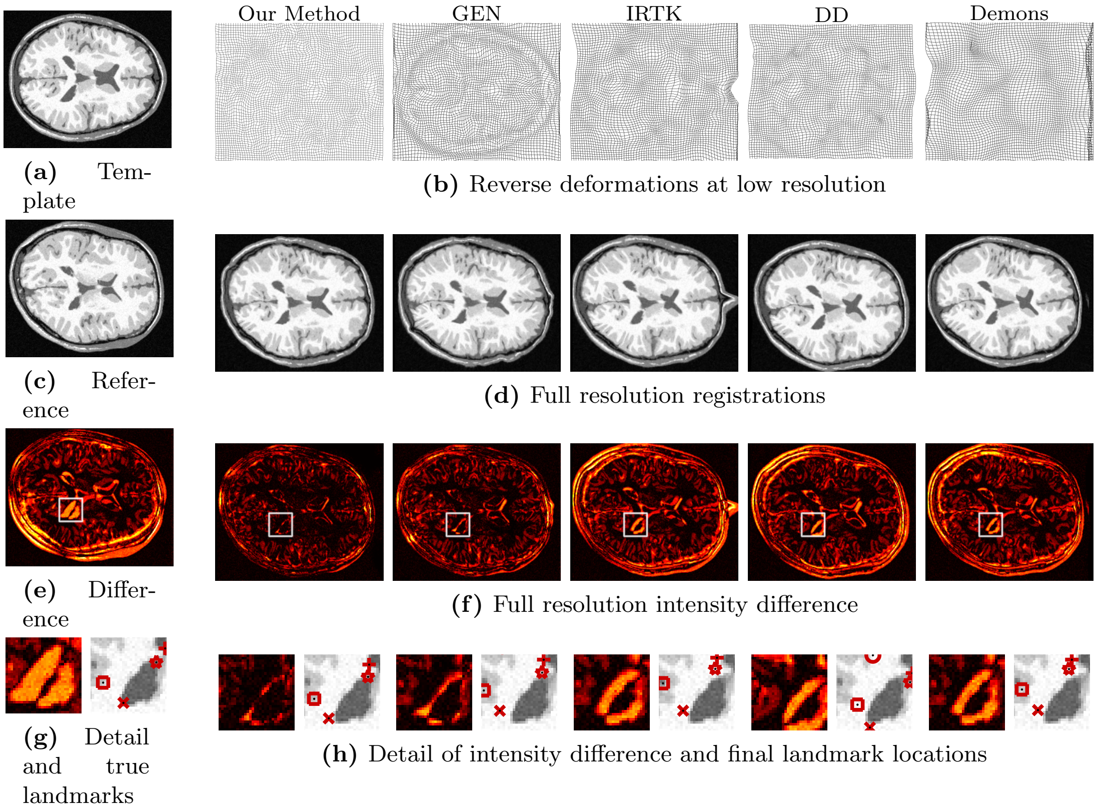
Having formulated the likelihood and prior constraints, we may estimate the maximum a posteriori using energy minimization. Specifically, the log posterior

$$\log p(\boldsymbol{Y}|\boldsymbol{X}, \theta) = \sum_{n=1}^{N} \log \sum_{m=1}^{M} e^{-\frac{1}{2}\left\| \frac{\boldsymbol{x}_n - \boldsymbol{y}_m}{H(\boldsymbol{y}_m^0)\sigma_0} \right\|^2} - \beta_{\mathrm{e}}\mathcal{E}(\boldsymbol{Y}) - \beta_{\mathrm{b}}\mathcal{B}(\boldsymbol{Y}) - \beta_{\mathrm{f}}\mathcal{F}(\boldsymbol{Y}) + C \tag{2.15}$$

can be maximized using simulated annealing over the scale parameter σ_0 [9]. We vary σ_0 between σ_{max} and σ_{min} which depend only on the size of the images and are set automatically. The solution for each iteration is found with an interior trust region method [38]. In practice our rigidity constraints have proven robust to large values for σ_{max}. Typically $\sigma_{min} = 0.5$, $\sigma_{max} = 6$, and 6 annealing iterations are needed to converge for 100×100 images. We set $t = 1$ and $c = 5$ in our folding prior $\mathcal{F}$ to allow faster convergence of each annealing iteration. With this smooth penalization, however, resulting registrations occasionally have some amount of folding of the registration. To address this, after each annealing iteration points that have folded over each other are merged together. Unfortunately, each iteration is still quite computationally expensive, requiring as much as forty-five minutes in our current unoptimized implementation. We envision a significant speed-up with an approximate linearization of the objective function.

2.4 Experimental results

We evaluate the accuracy of our model on human facial expressions and 2D slices of real brain MRIs. We compare the results with four characteristic automatic methods. Rueckert et al. [40] introduced an intensity-based approach that uses b-splines to smooth the deformation field which they released as part of their Image Registration Toolkit (IRTK). Thiron's well known Demons method [8], uses gradient information from the reference image to determine the amount of force the deforming points must exert. This work was later extended by Vercauteren et al. [10] to specifically model diffeomorphisims in a model termed Diffeomorphic Demons (DD). Finally, we compare our work to the generalized elastic net (GEN) model of Myronenko et al. [9] that uses a Gaussian mixture formulation similar to ours but with conventional smoothing priors. When possible, we use publicly available implementations of these algorithms with default parameters.



(a) Template

(b) Reverse deformations at low resolution

(c) Reference

(d) Full resolution registrations

(e) Difference

(f) Full resolution intensity difference

(g) Detail and true landmarks

(h) Detail of intensity difference and final landmark locations

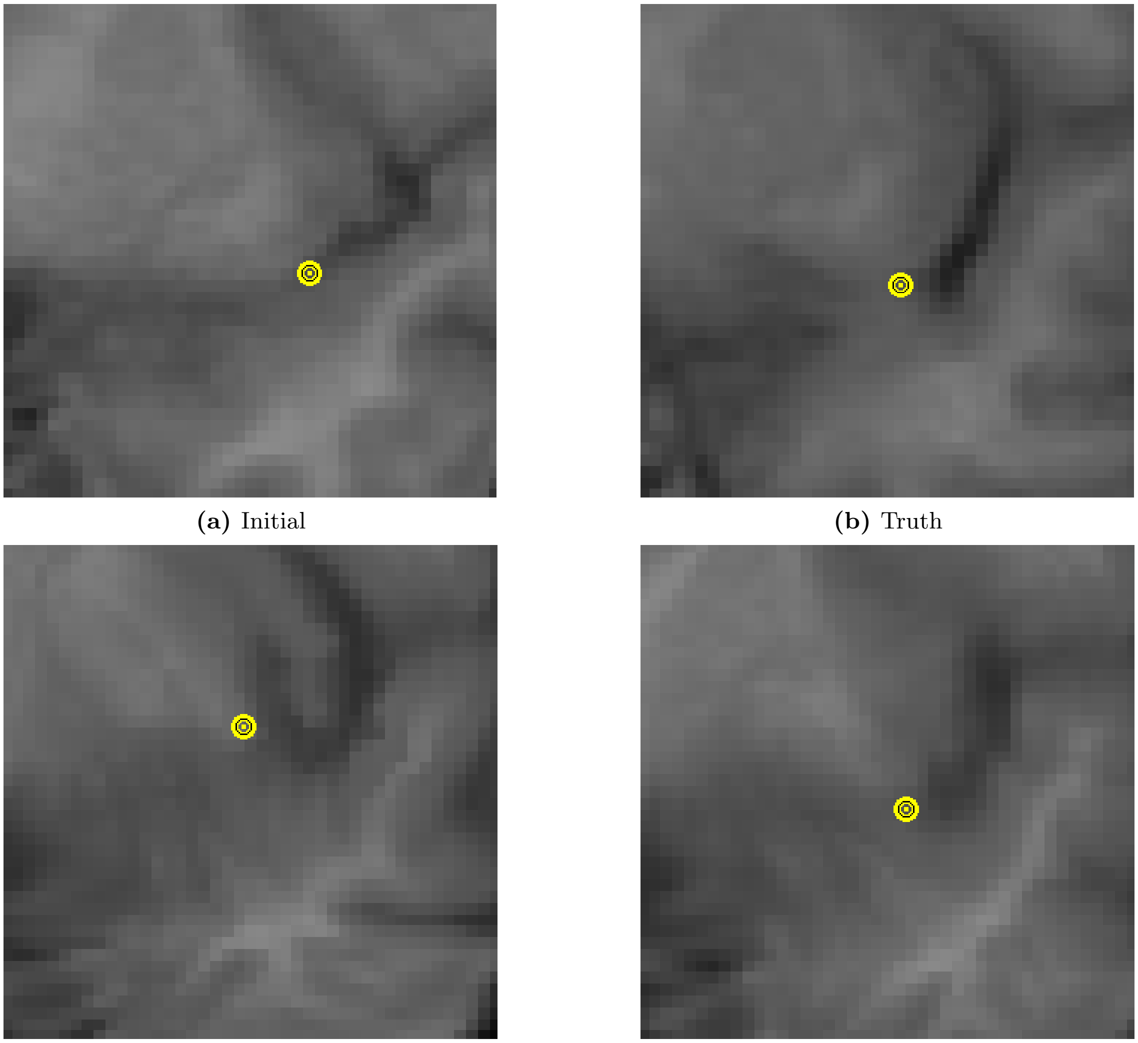
Figure 2.4: MRI registration comparison Template and reference images (from BrainWeb [39]) are scaled down and registration is performed with various methods. The resulting reverse deformation grid (b) is applied to the original template image (a). These registrations (d) are subtracted from the reference image (c). The error is then visualized (f) and compared with the difference of the original template and reference images (e). Increased brightness corresponds to larger error. Detailed inspections (h) of a region requiring a large transformation (g) show that our method results in the least error both in terms of the intensity difference and in the alignment accuracy of features. This example is labeled “Brain1” in Figure 2.7.

Past work use synthetic deformations to compare their results to ground-truth deformations. Synthetic deformations, however, are generated without regard for the physical structure of the image subject and therefore provide little information about real-world accuracy. Instead, we observe that an ideal registration should conform to the structural properties of the imaged subject. A deformation field embodying this characteristic should therefore maintain accuracy even when applied to a higher resolution image. At this increased resolution we may then compare the intensity error as well as the locations of manually labeled feature points to test sub-pixel accuracy. What may be termed “under-fitting” or “over-fitting” occurs when a deformation field appears well-suited at one resolution, but reveals significant inaccuracy at higher resolutions.

In Figure 2.4 we compare the results of our method on a subject from the BrainWeb database [39] with the other methods. A lower resolution version of the template image (2.4a) is registered to an equally down-sampled reference image (2.4c). The resulting inverse deformation fields (2.4b) show where each pixel in the resulting registration originated. The resulting high resolution registrations (2.4d), formed using a bilinearly interpolated inverse deformation field, are then compared with the reference image (2.4c) and the absolute difference is visualized as a heat map (2.4f) in which the brightness of the pixel increases as the error increases. Our approach produces a significantly improved registration, as evident by the greater amount of black (2.4f). Closer inspection (2.4h) shows the feature alignment accuracy of our method as evident by the close proximity of the feature points to key anatomical landmarks (2.4g). Here we also see that the error for this region is less than the interpolation scale (which is 3 in this case), revealing the degree of sub-pixel accuracy of our method. This example is labeled “Brain1” in Figure 2.7. Note that we achieve a minimum 29% decrease in overall intensity error, while achieving a 29% decrease in feature alignment error.

Figure 2.5 details the importance of shape preservation. When using a smoothing model, a landmark that was originally at the tip of a long feature looses this distinction and becomes embedded in a mass. With our method, the local geometry of the feature is preserved and a more accurate registration is achieved.

In Figure 2.6 we qualitatively compare the registrations a neutral face (2.6a) to a smil-

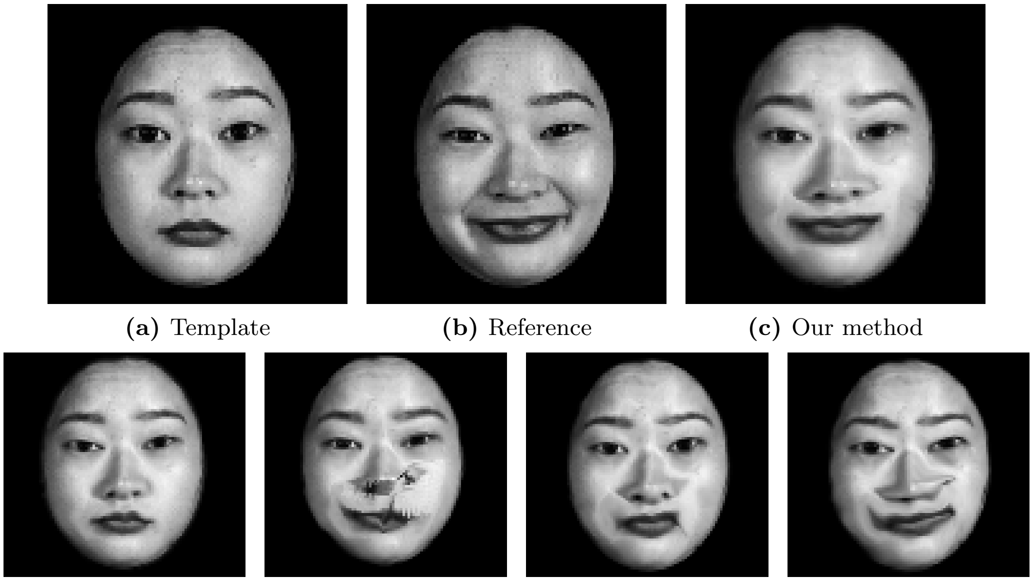


(a) Initial **(b)** Truth

(c) Smoothing **(d)** Ours

Figure 2.5: Landmark accuracy detail Our shape preserving priors (d) ensure that meaningful correspondences are made as compared to smoothing models [8] (c).

ing face (2.6b) from the Japanese Female Facial Expressions (JAFFE) database [41]. In analyzing the deformation fields we find that the key challenge in expression registration is the sudden appearance of dark regions that were not previously present. In this example the formation of a smile introduces dramatic changes in brightness in the cheeks as creases appear. This causes gross deformations to result in the other, less-structured models. Our method, on the other hand, achieves a much more accurate registration across the entire



(a) Template **(b)** Reference **(c)** Our method

(d) GEN **(e)** IRTK **(f)** DD **(g)** Demon

Figure 2.6: Facial expression registration Faces (from JAFFE [41]) present a particular challenge due to dramatic local deformations and intensity variations in corresponding regions like the creases of a smile. Our method outperforms past work by preserving the shape of the features as they deform. This example is "Face1" in the graphs of Figure 2.7.

face. Although it is not possible to recreate the creases without changing the intensity of the pixels, the shape of the lips and raise in cheeks are captured well.

In Figure 2.7 we quantitatively evaluate our results and compare them several other methods. For each dataset cluster the yellow column (labeled "None") shows the value of the error measures when no algorithm is used indicating the relative magnitude of the measures. In Figure 2.7a we compare the mean squared error in intensity $[0, 1]$ of the registration. The results show that our method results in an average of 32% less error than the next best method. In Figure 2.7b we compare the mean error of final landmark locations (in pixels) from the manually labeled ground-truth locations. For each image pair we annotate between 12 and 27 primary feature correspondences. Here our method achieves an average of 53% less error in feature alignment than the next best method with values consistently below the interpolation scale.

In Figure 2.8 we show a more detailed analysis of Face1, originally shown in Figure 2.6. The resulting reverse deformation grid (b) is applied to the original template image (a). The resulting registrations (e) are subtracted from the reference image (d). The error is then visualized (g) and compared with the error if no deformation is performed (f). Increased brightness corresponds to larger error. As was pointed out in the text, this example shows how the cheek highlights and creases are particularly challenging for the other methods whereas our method maintains facial features very well. In addition, many landmarks (k) are tracked and compared with the proper registration location (j). Note how the Diffeomorphic Demons method (second from right) has moved the right corner of the mouth so dramatically that it no longer represents a corner point in the final registration.

Figure 2.9 shows a detailed analysis of the results for Face3, with the same layout as Figure 2.8. In (k) we show the feature locations overlaid on the reference image. Here we see the key observation is found in the location of the registered mouth features – some methods register parts of the mouth to the chin crease. As was discussed in the text, many methods are unable to handle the ambiguity that arises as the mouth moves up and the chin crease forms. This facial expression results in a dark line both above and below the original mouth location. Other methods have no way to handle this ambiguity whereas our rigidity constraints ensure that the structure of the mouth is maintained during registration. This allows the portions of the mouth which are easily registered to help guide the rest of the lips as they deform.

Figure 2.10 shows a detailed analysis of the results for Face4, with the same layout as Figure 2.8. Here again we see that even with subtle deformations standard smoothing and diffeomorphic assumptions ultimately result in inadequate accuracy surrounding key features. Specifically, note in (e) how the other methods distort the shape of the nose, and the associated outline of error (g).

Figure 2.11, which shows Brain1 from the BrainWeb database [39], has the same layout as Figure 2.8. In (f) we see what appears to be a dramatic deformation. In (b) we compare the deformation grids and see that this case can actually be modeled with relatively small, local deformations. When comparing MRI slices one would expect that local features change their size and shape locally. Our model effectively models this whereas other models result in overly smooth and unrealistic deformation fields. Note how the top left portion of the ventricle has become almost completely occluded in the reference image. As is shown in the detail of Figure 2.3d, our method correctly models this as a crease whereas the other models have no way to register a feature that has disappeared.

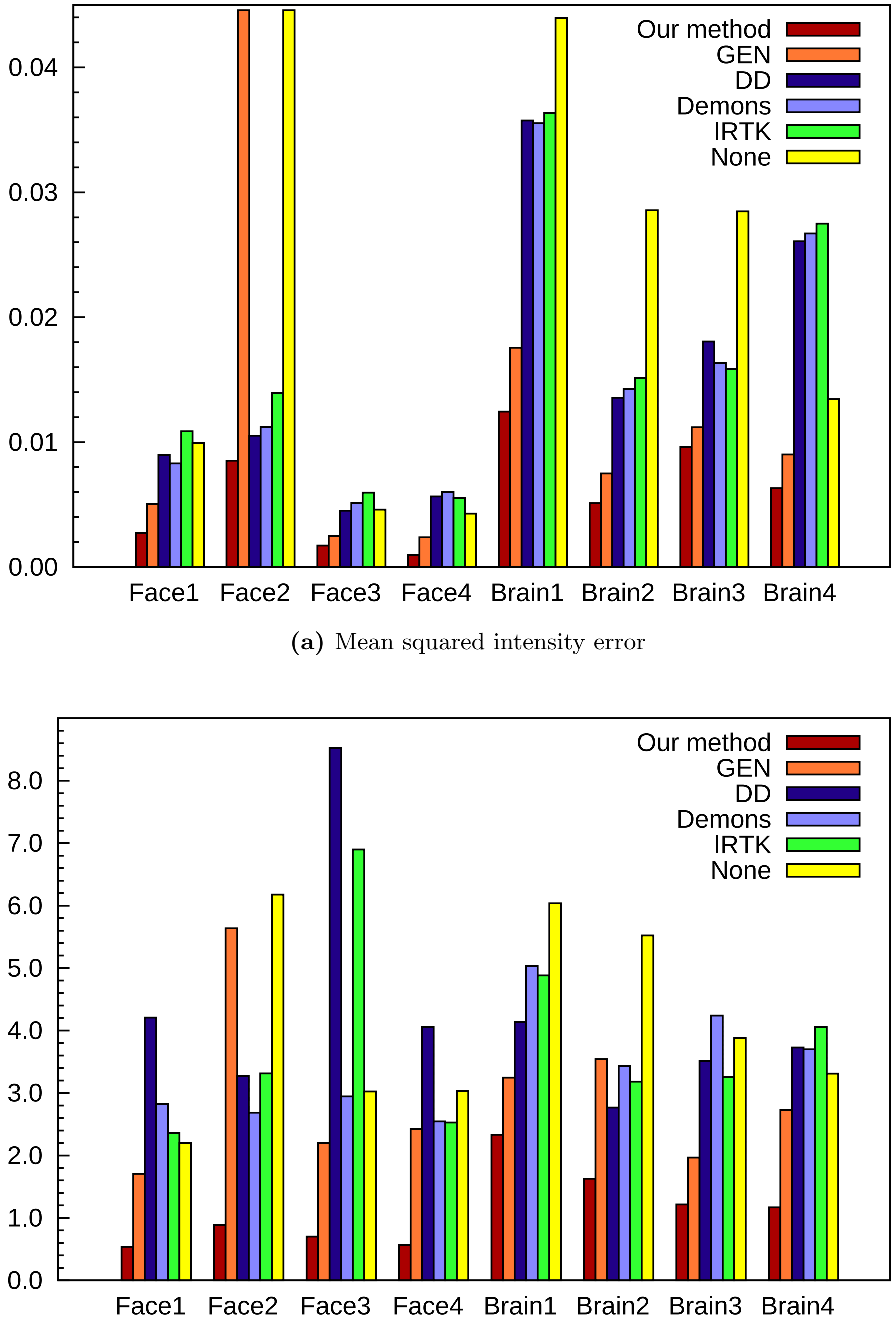
Figure 2.12, which shows Brain2 from the Internet Brain Segmentation Repository [42] has the same layout as Figure 2.8. In this example, as in many others, we can see how subtle the registration is using our method. Although some features move quite significantly, the global registration is almost completely unaffected by local deformations. In contrast, each of the other methods shows the effect of its own assumption of smoothness. The artifacts of such assumptions are evident in particular around the center ventricle where neighboring combinations of compression and expansion conflict and ultimately little improvement is made over the initial intensity difference (f).

Figure 2.13, which shows Brain3 from the Internet Brain Segmentation Repository [42], has the same layout as Figure 2.8. This is the most subtle example presented. In this case each of the methods has found a similar deformation. Even in cases like this one, however, our method shows an increase in accuracy both across the entire image and in the alignment of feature points.

Figure 2.14, which shows Brain4 from the BrainWeb database [39], has the same layout as Figure 2.8. Note how the top left ventricle has become so thin it is almost invisible in the reference image (d). The smoothness and diffeomorphic constraints of IRTK, Demons and Diffeomorphic Demons methods (right three columns) are unable to handle this compression (and other deformations). As can be seen in (b) the deformations produced are smooth, but dramatic. In fact we see that the overall intensity error actually increases. GEN (column 3), on the other hand, is essentially paralyzed and makes hardly any improvement. Our method is able to handle these deformations by explicitly modeling the non-diffeomorphic creases real-world cases like these exhibit.

2.5 Summary

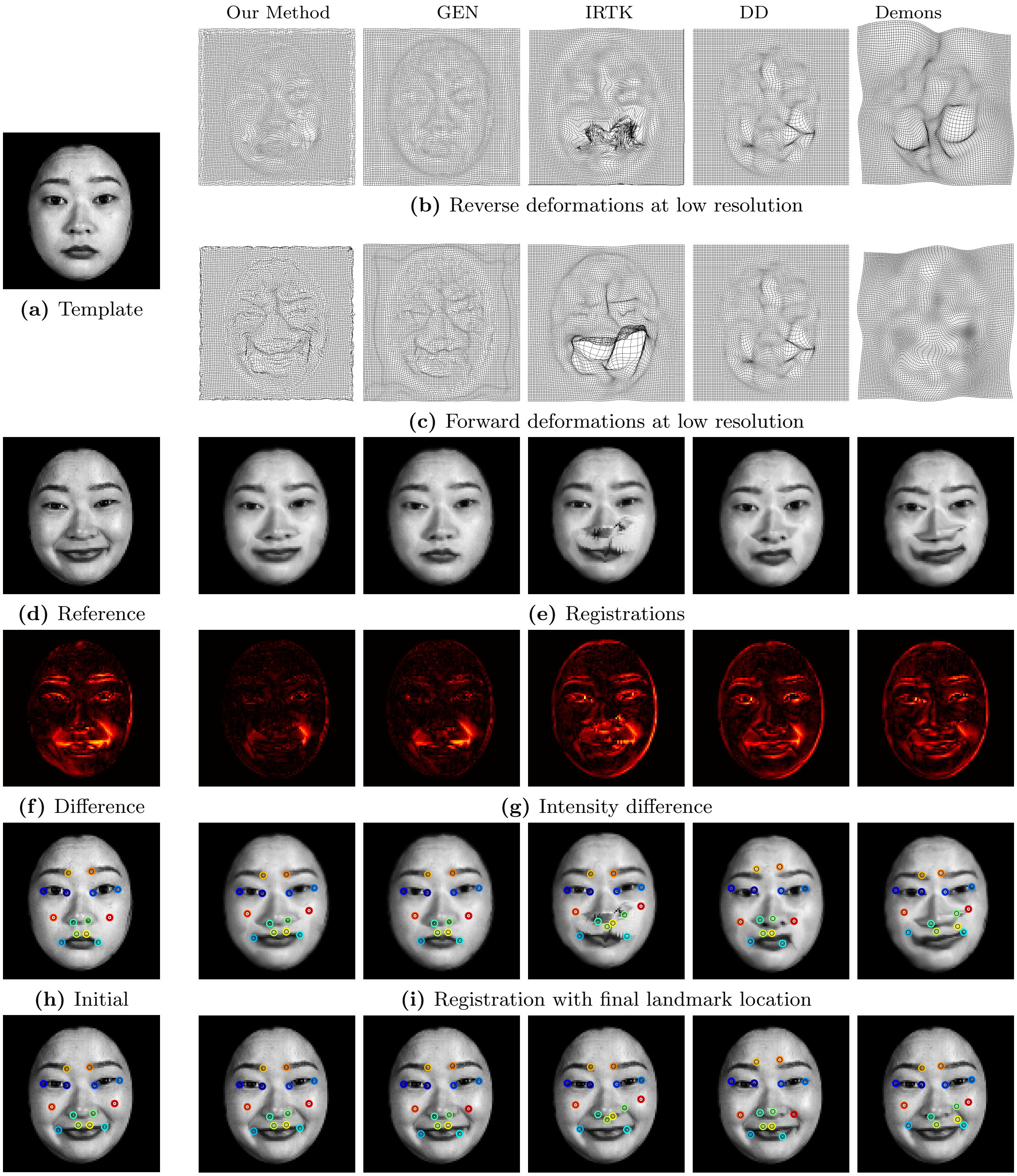
Our method demonstrates considerable accuracy that results from our key assumption – that the image as a membrane in 3D spatial-intensity space approximates the actual geometric surface of the subject and that by preserving its shape we can uncover the true deformation. Experimental results have shown that in many cases the assumption is valid and geometrically induced constraints increase accuracy dramatically. In particular, our method achieves higher accuracy in both the overall alignment and resulting feature correspondences. The resulting registrations exhibit a robustness to the common pitfalls of intensity-based registration techniques while maintaining particularly high accuracy for feature points automatically. This has strong implications in various applications where the accuracy of correspondences is particularly important.



(a) Mean squared intensity error

(b) Mean landmark error (in pixels)

Figure 2.7: Comparison of all results Mean squared intensity error for full images (a) and mean landmark distance error for suites of manually annotated features (b). In both graphs the yellow (right-most) bar of each grouping indicates the amount of error if no registration is performed. Our method consistently outperforms every other benchmark method.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Registrations

(f) Difference

(g) Intensity difference

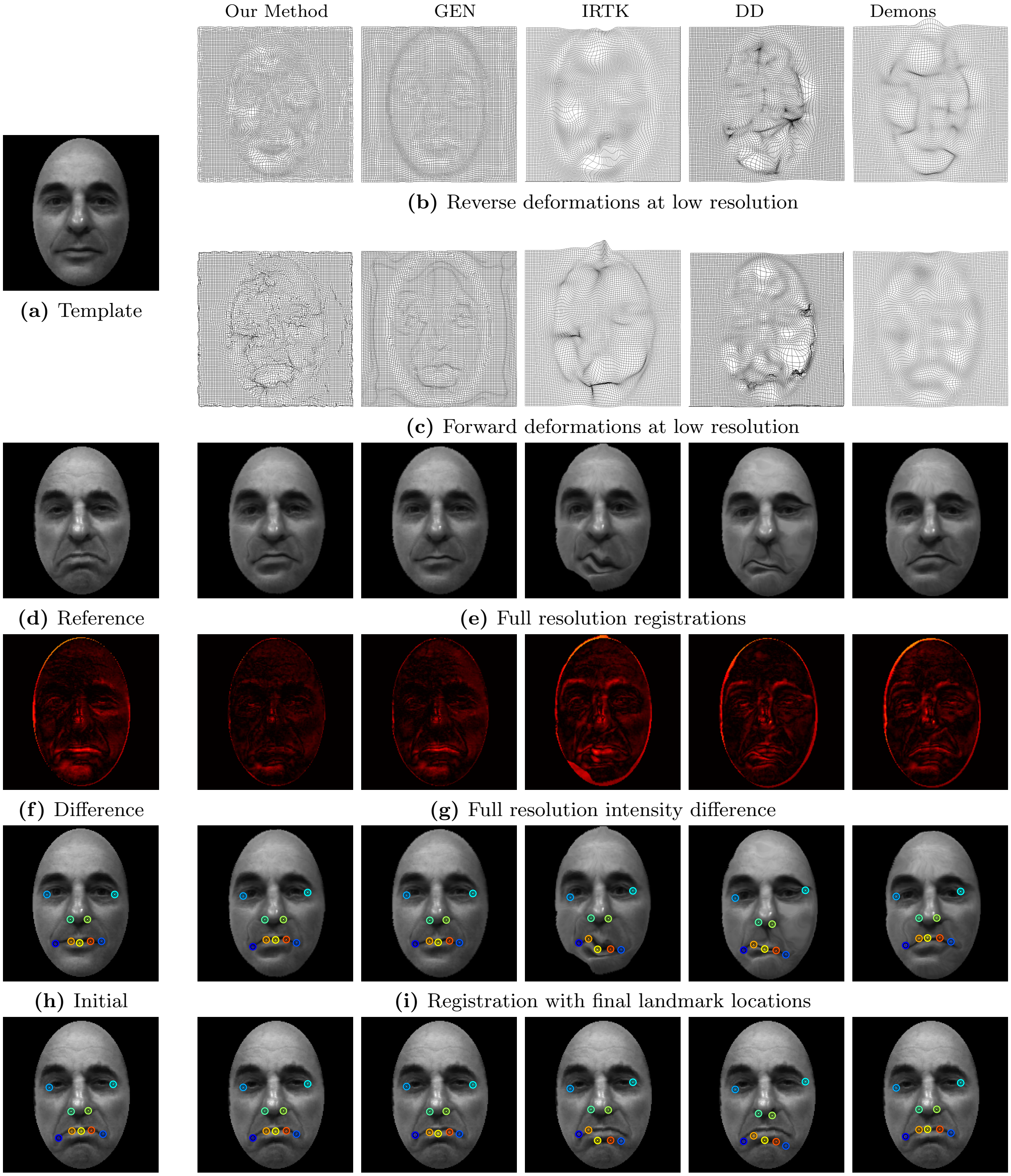
(h) Initial

(i) Registration with final landmark location

(j) True

(k) Final landmark locations shown on reference image

Figure 2.8: Detailed results for Face1 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

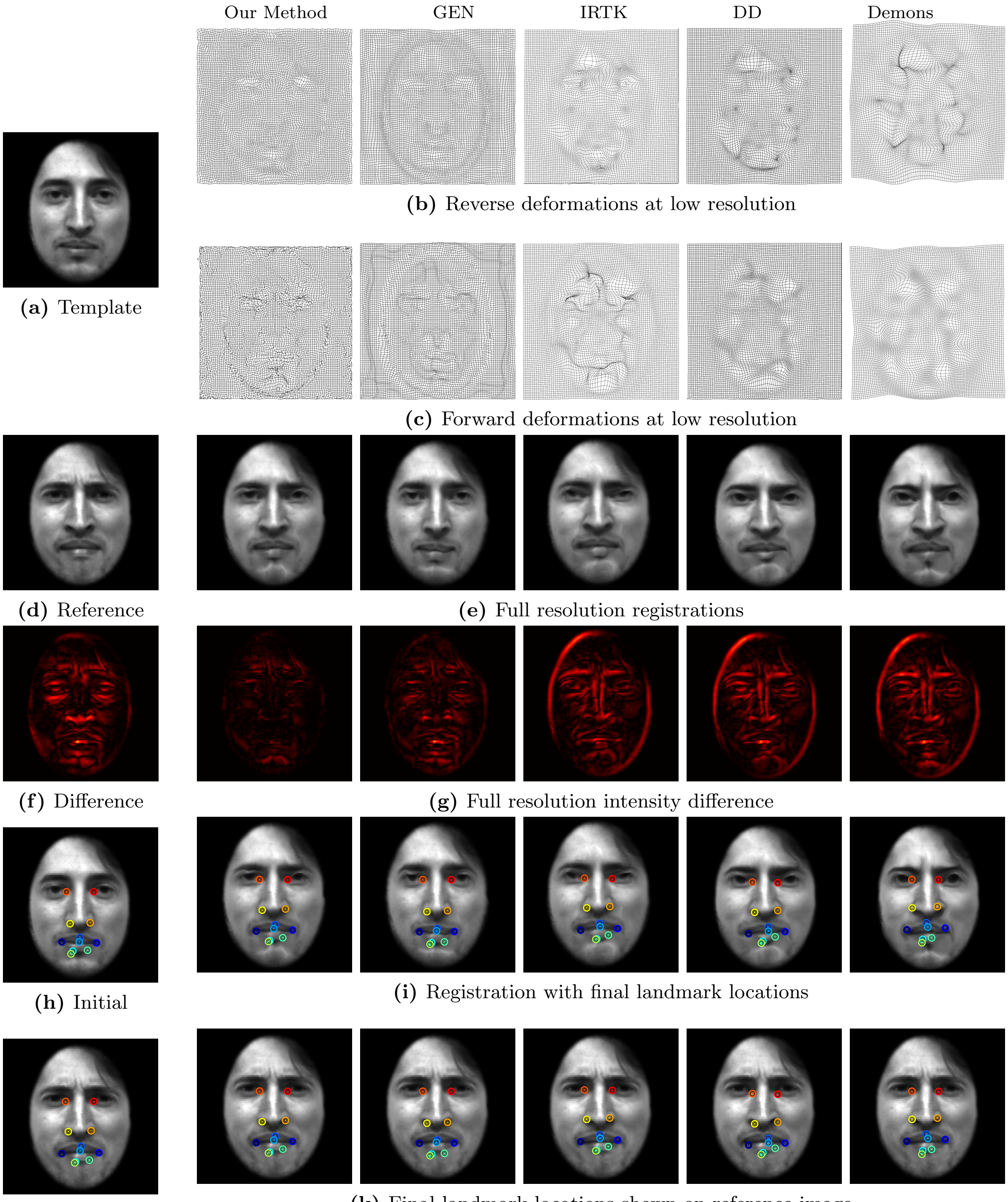
(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final landmark locations shown on reference image

Figure 2.9: Detailed results for Face3 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

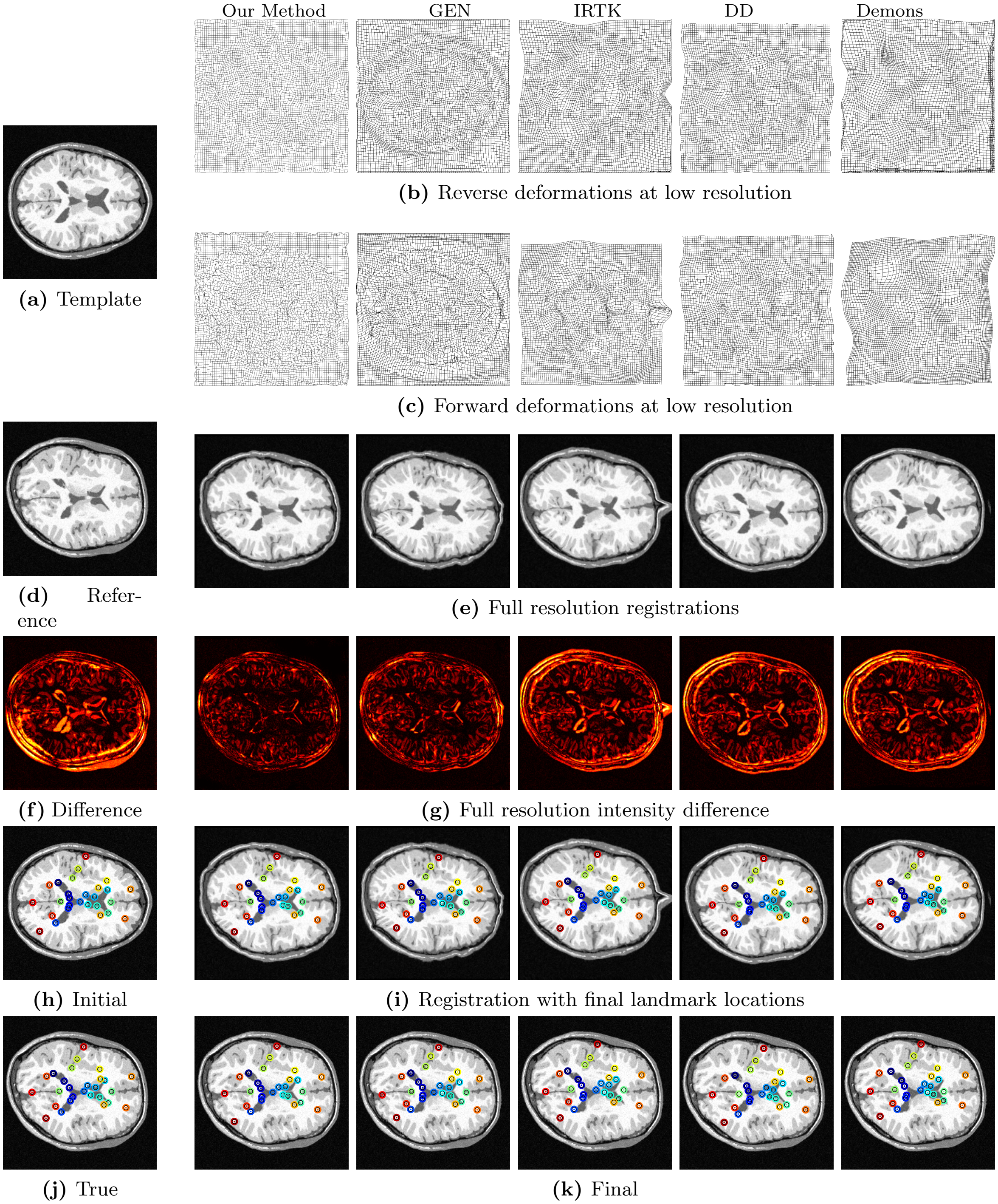
(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final landmark locations shown on reference image

Figure 2.10: Detailed results for Face4 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

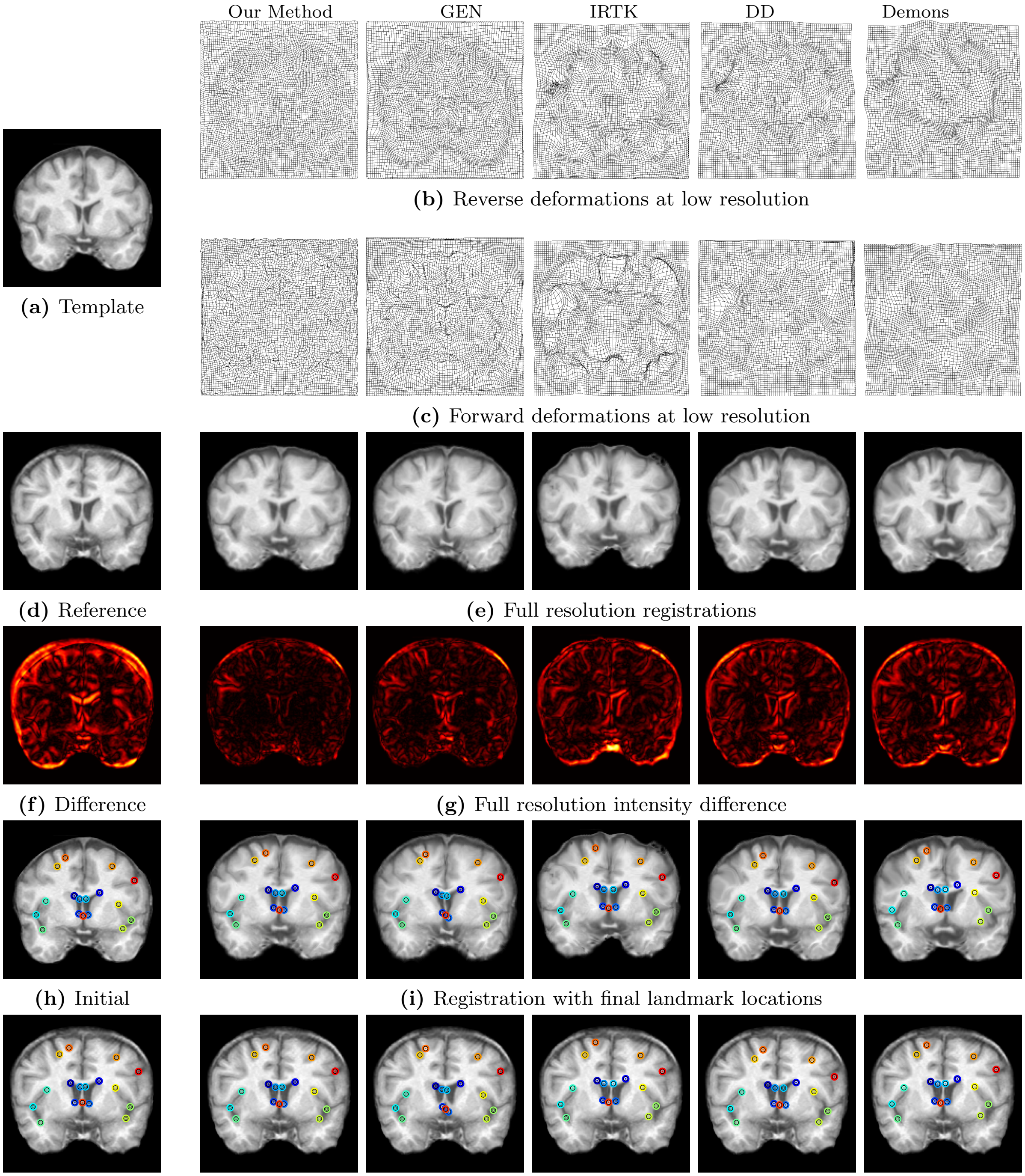
(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final

Figure 2.11: Detailed results for Brain1 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

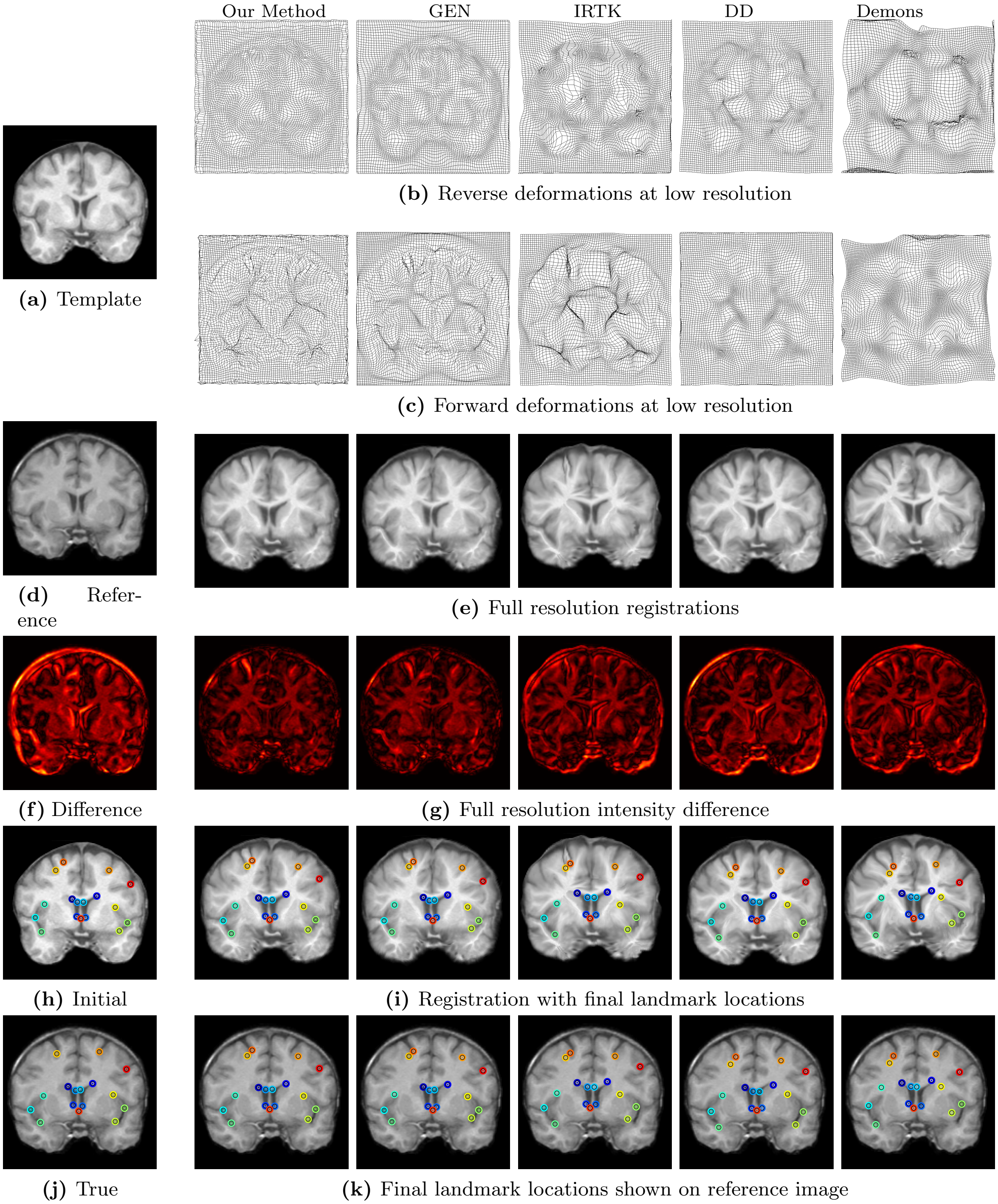
(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final landmark locations shown on reference image

Figure 2.12: Detailed results for Brain2 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

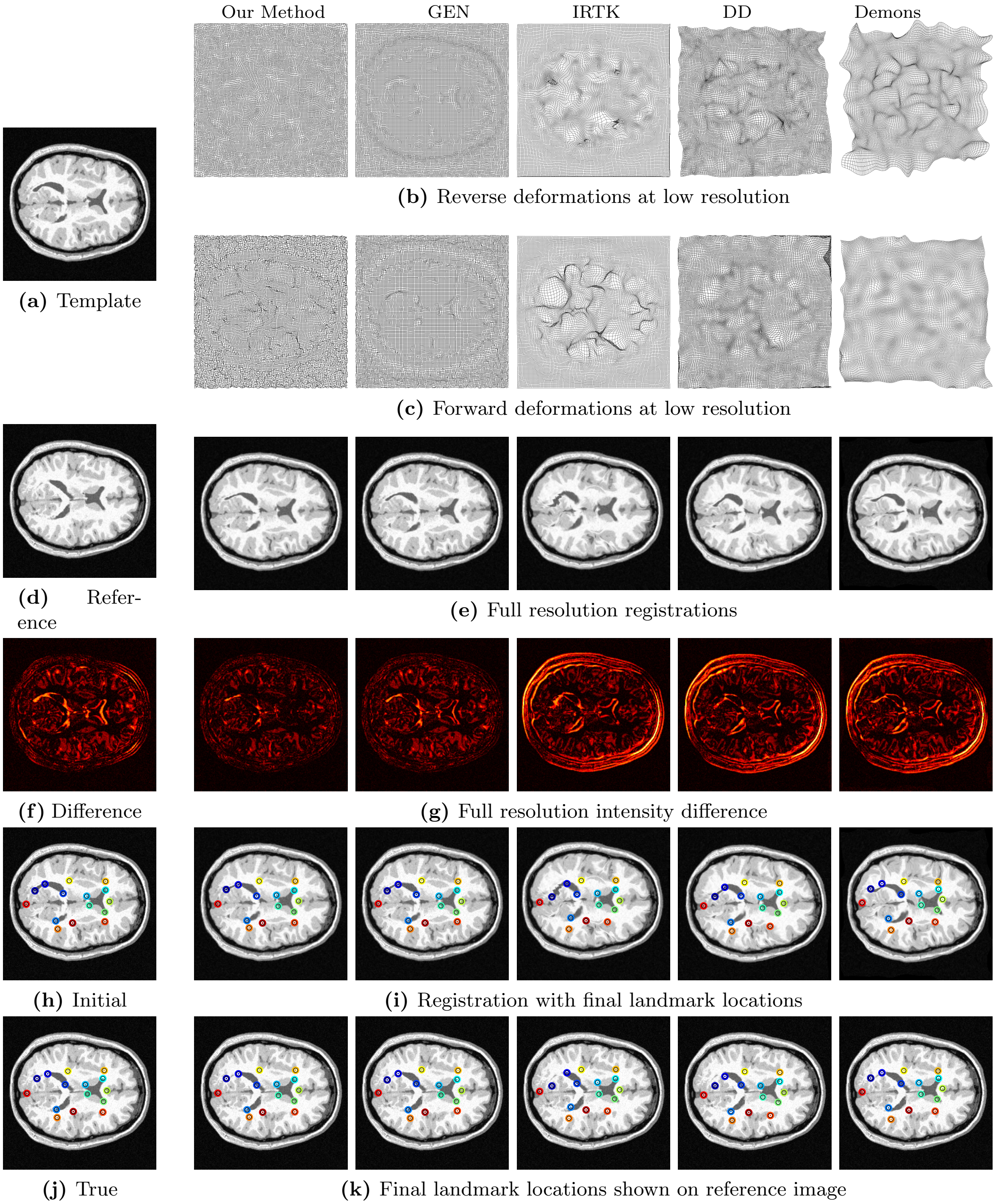
(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final landmark locations shown on reference image

Figure 2.13: Detailed results for Brain3 See text for discussion.



(a) Template

(b) Reverse deformations at low resolution

(c) Forward deformations at low resolution

(d) Reference

(e) Full resolution registrations

(f) Difference

(g) Full resolution intensity difference

(h) Initial

(i) Registration with final landmark locations

(j) True

(k) Final landmark locations shown on reference image

Figure 2.14: Detailed results for Brain 4 See text for discussion

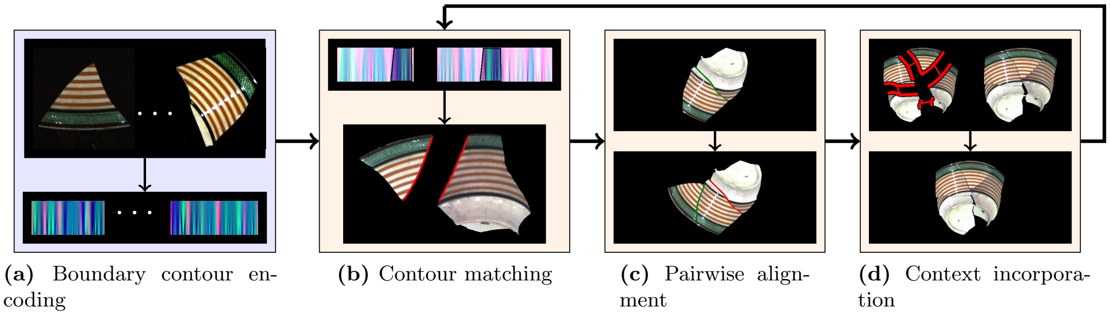
Chapter 3: Fragmented geometry recovery

3.1 Introduction

In this chapter we investigate the problem of broken geometry. Reassembly is a geometry recovery problem in which we are given a set of incomplete observations in the form of 3D fragments, that, when aligned correctly, form a whole object. This problem, which may be thought of as a 3D jigsaw puzzle, has important applications in forensics, archeology, and medical imaging. In addition to easing this time consuming process, by working on virtual representations of the objects, we free ourselves from the danger of further damaging the objects, or destroying fingerprints or other subtle but important features.

Just as one would try to mend a broken vase by hand, the basic approach is to incrementally locate matching fragments, and align (glue) them together. The majority of past work has focused on thick objects, such as tile-work, or sculptures because the fractures expose unique "break-surfaces." Because these surfaces will have matching geometric features (peaks and valleys), they are a reliable way to find matching fragments. Once two fragments are determined to match, the entire set of break-surface points can then be used to accurately align the fragments. For thin objects, such as a skull, or ceramic pottery, both the matching and aligning problems become much more challenging. Instead of sharing oppositely shaped break-surfaces, fragments of thin objects share only a minute strip in common.

To account for the apparent lack of information, past work has made restrictive assumptions about the shape, or painted texture of the object. We avoid such assumptions by extracting and exploiting the geometric and photometric structure encoded along the


(a) Boundary contour encoding

(b) Contour matching

(c) Pairwise alignment

(d) Context incorporation

Figure 3.1: System overview. We present a three-step method to reassemble real-world objects of arbitrary shape using only the boundary of each fragment's external surface. To enable rapid matching and accurate alignment, we first encode the scale variability of each fragment's shape and color as a multi-channel 2D image (a). 1) Using this novel scale-space representation, we quickly identify matching boundary regions (b). 2) We then align the fragments by optimizing the geometric continuity across them (c). accuracy (d). 3) Finally, we incorporate the additional contextual information each match provides to ensure global accuracy and to improve subsequent matches (d).

fragment boundaries. In this chapter, we present our three-step method that reassembles objects using only the fragment boundary contours with minimal user interaction.

Figure 3.1 outlines our approach. (a) First, we preprocess each fragment to encode the scale variability of its boundary contour as a multi-channel 2D image. (b) We then identify matching sub-contours using a novel image registration method based on these scale-space boundary contour representations. (c) Next, we estimate the transformation to align the fragments using a least squares formulation that minimizes the distance between the adjoining regions while simultaneously maximizing the resulting geometric continuity across them. (d) The configuration of the fragments as they are incrementally matched and aligned form a graph structure. By identifying cycles in this graph, we detect subsets of fragments whose alignments are dependent on each other. When a cycle is formed, we jointly re-optimize the alignments of the constituent fragments to ensure a globally optimal configuration, and improve subsequent matches.

We use ceramic pottery as the driving example, and validate our method on several recently excavated real-world historic artifacts. The results show that our method allows for accurate reassemblies to be achieved with minimal human interaction, even when many fragments are missing.

3.2 Related work

When reassembling an object from its fragmented parts, the basic approach is to iterate between finding matching fragments and bringing them into alignment. Reassembling thin objects is particularly challenging because matching fragments share only their boundaries in common. In other words, their “break-surfaces” are not surfaces at all, but contours. On their own, these contours provide insufficient information to reliably match candidate fragments since many fragments have similar shapes. This lack of information also makes aligning matching fragments challenging. We cannot simply place the two matching boundary contours next to each other since this does not guarantee that the fragment surfaces themselves will reform the original unbroken shape.

To address the ill-posed problems of finding matching fragments and aligning them, a common approach is to restrict the class of objects that may be reassembled by imposing a global model. Some authors [13–15] assume the object is axially symmetric (a vase or bowl). The shape of the fragment can then be compared to the profile contour of the object model to help determine its proper location and orientation. This effectively reduces the problem to a sort of curved 2D puzzle. To determine the profile contour of the object, the authors also assume that the fragments themselves are able to reliably convey this information. If the fragments are too small, the profile contour may not be estimated. We avoid relying on simple object models, or assuming sufficiently large fragment sizes by exploiting the geometric continuity that must exist across matching fragments.

Even when a global model is used to simplify the reassembly process, methods that rely on pairwise alignments necessarily suffer from error accumulation. This effect can even be seen when the object has only three pieces. Once the first two pieces have been matched, the space for the third piece may already be too small or too large. To address this, some authors [14, 43] delay the alignment phase until clusters of three matching fragments have been found. While this does improve alignment accuracy, it is still a local optimization and therefore prone to error accumulation.

In range image registration, where pairwise alignment is much more reliable, accumulated error may be evenly dispersed to finalize a reassembly. Once a series of overlapping fragments is found that connects back to itself, Sharp et al. [44] then divide up the resulting gap or overlap over the whole set of fragments. Pulli [45] uses the point correspondences of the pairwise alignments as soft constraints on a final global alignment. In object reassembly, however, the pairwise alignments are not as reliable. In addition to improving the overall reassembly accuracy, each incremental match provides important contextual clues that can be used to improve subsequent matches. By leveraging the context provided by past matches, we determine the placement of relatively nondescript fragments, and ensure global accuracy.

Sağiroğlu and Erçil [16] focus on the fragments' painted texture. If the object is sufficient patterned, they use texture synthesis and inpainting methods to expand each fragment's surface texture. This can be thought of as generating overlapping surfaces which may then be aligned using standard texture matching approaches. We also use the texture painted onto the surface, but only as part of our representation of the boundary contour and do not assume any a priori knowledge of the pattern or characteristics of the texture.

Some authors focus specifically on the problem of finding matching boundary contours of thin fragments using only their 3D geometry. Although the contours are three dimensional, Wolfson [46] showed that finding matching sub-contours can be recast as a one-dimensional string matching problem. This dimensionality reduction is possible because contours can be uniquely expressed in terms of their rotationally and translationally invariant geometric characteristics: curvature and torsion. By encoding contours in this way, matching regions may be detected as matching series of curvature and torsion tuples using the longest common substring algorithm.

Unfortunately, recasting 3D contour matching as 1D string matching is highly sensitive to noise, chipping and other realities since it relies on the calculation of the contour's third derivative. To address this, including scale in the matching process has been quite successful. Large scale detail can be used to estimate matches, while finer scale detail can be used to refine and validate the match. The standard approach to incorporate scale is to incrementally smooth the 3D points of the contour, recalculating the curvature and torsion values at each degree of smoothness [46–50]. Although incremental smoothing highlights features of prominence and mitigates the effect of noise, smoothing the *geometry* of the contour introduces shapes that are dramatically different from the original. Our approach is more faithful to the notion of scale. By smoothing the geometric *characteristics* of the contour we induce an effect similar to moving away from the object. In addition to this, we incorporate the photometric characteristics of the boundary contour. This additional information helps inform the contour matching process, and is particularly useful when many boundary contours share similar geometry.

3.3 Boundary contour representation

The first step in our method is to identify which fragments are most likely to align, and where their boundaries match. To do so quickly and accurately, we leverage the scale

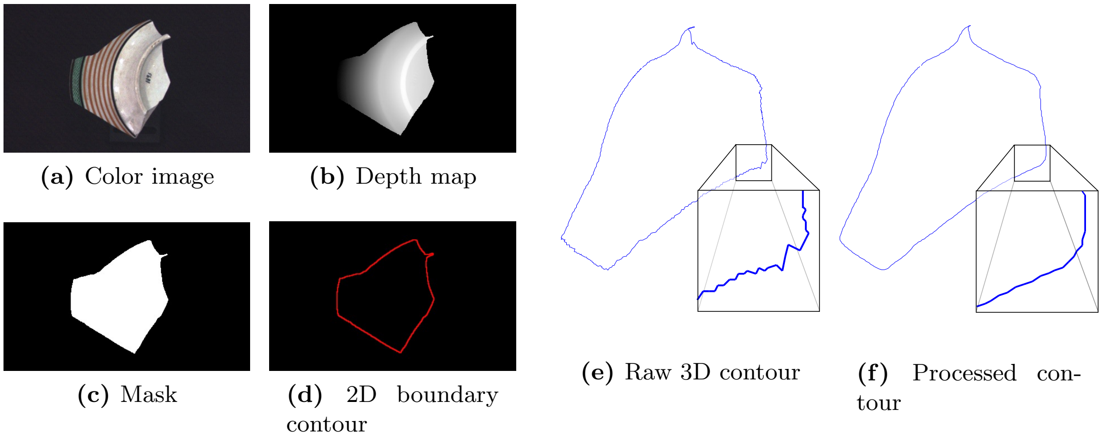

(a) Color image
(b) Depth map
(c) Mask
(d) 2D boundary contour
(e) Raw 3D contour
(f) Processed contour

Figure 3.2: Boundary contour extraction. We start with a single 3D scan of each fragment. Each scan is comprised of a color image (a) and a corresponding grid of 3D point locations shown here as a depth map (b). Using depth and color discontinuities, we generate a mask (c) from which we extract the 3D boundary contour (e). We then smooth and sub-sample the contour to minimize noise and to ensure a uniform arc-length parameterization (f). This enables reliable derivative calculations without noticeably changing the shape of the fragment.

variability of each fragment's boundary contour. A coarse scale representation, which is robust to noise and subtle detail, may be used to quickly estimate potential matches, while finer scale detail may be used to verify and fine-tune the estimated matches. This graduated relationship naturally lends itself to a hierarchical encoding. To that end, we build a multi-channel image representation that encodes the scale variability of each fragment's shape and color.

3.3.1 Boundary Extraction

As shown in Figure 3.2, we start by acquiring geometric and photometric information about each fragment using a range sensor; specifically a light-stripe range scanner Canon VIVID 910. Without loss of generality, we assume the entire surface of each fragment may be viewed from a single viewpoint. Although this assumption holds true for the various objects in our datasets, if it were not the case, multiple scans could be used to form the exterior surface of the fragment. Each scan is comprised of a 640×480 color image (Figure 3.2a), as well as three-dimensional coordinates for each of the image points (represented in Figure 3.2b as a depth map). In most cases we use depth and color discontinuities to automatically isolate the fragment in the scan. The result is a 2D mask like the one shown in Figure 3.2c. Occasionally, part of the thin break-surface is visible and manual intervention is required to correct the mask.

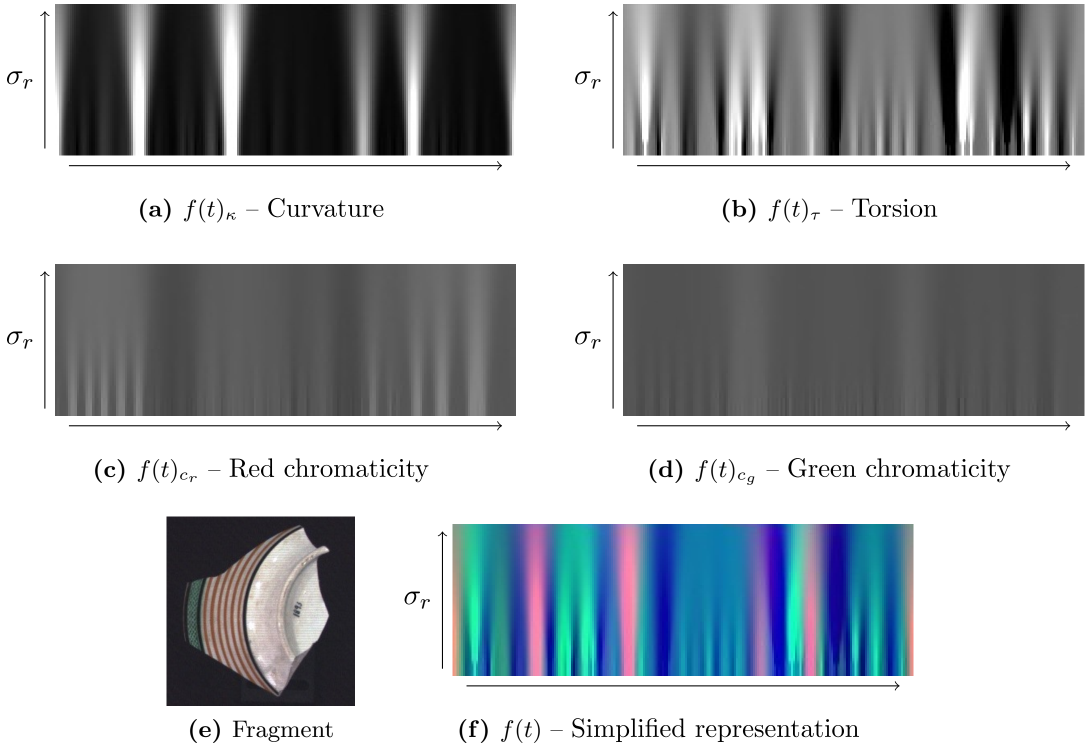
Using Suzuki's border following method [51], we extract the 2D border of the mask (shown in Figure 3.2d). We then extract the 3D location of each point along the border from the range data. This contour (shown in Figure 3.2e) is noisy and its sampling is too dense to allow for reliable analysis. To address this, we smooth the contour slightly and sub-sample the points using an iterative technique described by Leitao and Stolfi [48] which ensures that the points along the resulting contour are uniformly spaced. The processed boundary contour (Figure 3.2f) exhibits a necessary degree of smoothness to reliably evaluate while remaining true to the original shape. Once the geometry of the 3D contour has been processed, we project the contour back to the image plane. Here we extract the color of each contour point using bilinear interpolation.

3.3.2 Boundary Representation

We describe each fragment's boundary contour as a cyclic string $f(t)$ of four-valued feature vectors

$$f(t) = \{(\kappa, \tau, c_r, c_g)_{t \bmod n}\} \ , \tag{3.1}$$

where n is the number of samples along the contour and t is the sample index. The first two values, κ and τ, are curvature and torsion. These two values encode the rotationally and



(a) $f(t)_\kappa$ – Curvature **(b)** $f(t)_\tau$ – Torsion

(c) $f(t)_{c_r}$ – Red chromaticity **(d)** $f(t)_{c_g}$ – Green chromaticity

(e) Fragment **(f)** $f(t)$ – Simplified representation

Figure 3.3: Boundary contour representation. We encode the scale variability of each fragment's shape and color. The first two channels (curvature and torsion) encode the shape and the second two (red and green chromaticity) encode the color. The raw values are embedded at the base of the image. Moving up the image corresponds to increased smoothing of the base row with a Gaussian kernel of standard deviation σ_r. Shown at the bottom right is the compact visualization used throughout this paper. Here, the red channel is curvature, green is torsion, and blue is the intensity of each contour point.

translationally invariant geometry of the 3D contour. The second two, c_r and c_g, are the red and green chromaticity; they encode the appearance of the contour. These values help locate and refine matches, particularly when the contour geometry is relatively featureless. We use chromaticity because it provides a measure of invariance to illumination conditions, which is an important consideration in many domains. Also, observe that the string $f(\cdot)$ is periodic, i.e., $f(t+n) = f(t)$. This captures the cyclic nature of the contour.

The curvature and torsion (κ and τ) of a 3D contour $\boldsymbol{\lambda}(t) = (x(t), y(t), z(t))$ can be defined in terms of the derivatives of the three coordinates [52] as,

$$\kappa = \frac{\|\ddot{\boldsymbol{\lambda}} \times \dot{\boldsymbol{\lambda}}\|}{\|\dot{\boldsymbol{\lambda}}\|^3} = \frac{\sqrt{A^2 + B^2 + C^2}}{(\dot{\boldsymbol{x}}^2 + \dot{\boldsymbol{y}}^2 + \dot{\boldsymbol{z}}^2)^{3/2}} \tag{3.2}$$

and

$$\tau = \frac{(\dot{\boldsymbol{\lambda}} \times \ddot{\boldsymbol{\lambda}}) \cdot \dddot{\boldsymbol{\lambda}}}{\|\dot{\boldsymbol{\lambda}} \times \ddot{\boldsymbol{\lambda}}\|^2} = \frac{\begin{vmatrix} \dot{\boldsymbol{x}} & \dot{\boldsymbol{y}} & \dot{\boldsymbol{z}} \\ \ddot{\boldsymbol{x}} & \ddot{\boldsymbol{y}} & \ddot{\boldsymbol{z}} \\ \dddot{\boldsymbol{x}} & \dddot{\boldsymbol{y}} & \dddot{\boldsymbol{z}} \end{vmatrix}}{A^2 + B^2 + C^2} \,, \tag{3.3}$$

where

$$A = \begin{vmatrix} \dot{\boldsymbol{y}} & \dot{\boldsymbol{z}} \\ \ddot{\boldsymbol{y}} & \ddot{\boldsymbol{z}} \end{vmatrix} \,,\ B = \begin{vmatrix} \dot{\boldsymbol{z}} & \dot{\boldsymbol{x}} \\ \ddot{\boldsymbol{z}} & \ddot{\boldsymbol{x}} \end{vmatrix} \,,\ C = \begin{vmatrix} \dot{\boldsymbol{x}} & \dot{\boldsymbol{y}} \\ \ddot{\boldsymbol{x}} & \ddot{\boldsymbol{y}} \end{vmatrix} \,. \tag{3.4}$$

Here $\|\Box\|$ and $|\Box|$ denote the $L2$ norm and matrix determinant, respectively. $\dot{\Box}$, $\ddot{\Box}$, and $\dddot{\Box}$ denote the first-, second-, and third-order derivatives with respect to the arc length t, i.e., $\frac{\partial}{\partial t}$, $\frac{\partial}{\partial t^2}$ and $\frac{\partial}{\partial t^3}$. We compute these using central numerical differentiation.

These geometric values are used frequently in shape description applications because they provide a unique, rotationally and translationally invariant representation of the contour. We extend the descriptive formulation beyond past work by additionally encoding the photometric properties of the contour. After white-balancing the color image, we then encode the color of each contour point using its red and green chromaticity values (c_r and c_g respectively). These are computed by simply dividing the color channels by the total intensity of the point. Note that including the blue channel would be redundant since the three values sum to one.

3.3.3 Encoding Scale

Although this string of values $f(t)$ accurately describes the boundary contour, it encodes small-scale detail and noise that may not precisely align with the matching sub-contour description of an adjoining fragment. In order to reliably locate matching boundary sub-contours, we exploit the scale variability of their geometry and photometry. We use the coarse scale representation, which is robust to noise and subtle detail, to quickly estimate potential matches, and the finer scale detail to verify and fine-tune them.

Past authors [46–50] typically encode scale by incrementally smoothing the geometry of the contours themselves. This approach, however, introduces shapes that are dramatically different from the original. In order to maintain the authenticity of the underlying geometry and photometry of the contour our approach is to iteratively smooth the string $f(t)$ itself. We then store the smoothed values as rows of a multi-channel image $\mathbf{S}$, like the one shown in Figure 3.3. For clarity, we have shown each channel separately. Left-to-right, top-to-bottom they are curvature, torsion, red and green chromaticity. The lowest row of pixels $\mathbf{S}_0$ encodes the smallest scale detail, i.e., $\boldsymbol{S}_0 = f(t)$. Moving up the image to row r corresponds to a coarser scale. Each increasing scale is calculated using circular convolution of the base row with a Gaussian smoothing kernel $\mathcal{N}(\sigma_r)$ of standard deviation σ_r proportional to r,

$$\boldsymbol{S}_r = \mathcal{N}(\sigma_r) \circledast \boldsymbol{S}_0 \ , \tag{3.5}$$

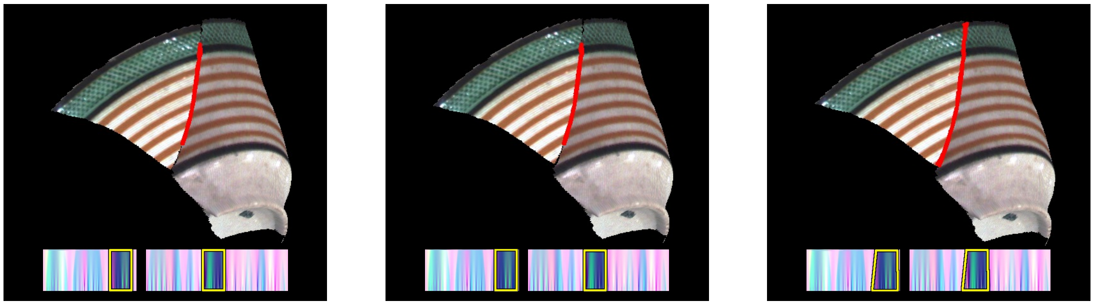
where $\circledast$ is the circular convolution operator (convolution with periodic a boundary). In our case, we let $\sigma_r = 0.05r$ and build images with 100 rows. Increasing the number of rows, and consequently the maximum value for σ, will increase the number of false positives as discerning detail becomes smoothed away. Decreasing the number of rows, on the other hand, will lead to false negatives and ultimately, incomplete reassemblies. We therefore chose this value empirically to minimize the false positives without allowing for many false negatives.

The resulting representation is now a 4-channel 2D image that encodes the scale variability of the boundary contour's geometry and photometry. In Figure 3.3, observe how the five sharp corners of the fragment (e) have become bright regions in the curvature channel (a), and how the banded painting yields a similarly striped pattern in the red chromaticity channel (c). Note that as torsion (b) is a signed value, in this channel, a value of 0 is given an intensity of 0.5. Although the image has four channels, we use a more compact rendering throughout this paper. As shown in (f), we replace the two chromaticity channels with a single channel computed as the intensity. This channel is colored blue; curvature is shown in red, and torsion is green.

This 2D formulation is particularly appealing because it describes the photometric and geometric properties of the contour under all scales at once. It is this characteristic that we leverage when solving the matching problem.

3.4 Matching boundary contours

By encoding the scale variability of each boundary contour's shape and color as a multichannel image, the problem of finding matching sub-contours is now akin to partial image registration. Specifically, matching image regions will correspond to matching sub-contours. Note that because both contours are encoded in counterclockwise ordering, one image must first be horizontally flipped. Although chromaticity has a standard range of $[0, 1]$, curvature and torsion are unbounded. To introduce a degree of comparability across all channels, we limit each range by scaling it. Figure 3.4 outlines our three step matching method.



(a) We first find matching sub-regions using the longest-common substring algorithm at the coarsest scale.

(b) We then refine the match using all scales with normalized cross-correlation image registration.

(c) Finally, we expand the base of the matching region to account for the fan-out of non-matching end-points.

Figure 3.4: Boundary contour matching. We extract matching boundary contours (red) as matching trapezoidal regions (highlighted boxes) of our scale-space images. Matches are estimated (a), refined (b), and optimized (c) quickly using this scale-space representation. Note how the refinement (b) has shifted the matching region right in the scale-space image (clockwise along the boundary), and how the fan-out optimization (c) has expanded the base to the full width of the matching boundary regions.

a) Longest common sub-contour Similar to past work [43,49,50], we begin our search with the coarsest scale, i.e., the top row. Here we perform a longest common substring analysis with two modifications. First, a pair of four-valued feature vectors $r_0^{\boldsymbol{A}}(p)$ and $r_0^{\boldsymbol{B}}(q)$ is considered to be matching if each of their values differ by less than a threshold. Secondly, we allow matches to wrap around the images. The table produced by the dynamic programming algorithm is then traversed, and matches are inserted into a queue where priority is given based on the length of the match. Figure 3.4a, shows the top match for the two fragments shown.

b) 2D image registration refinement Since this coarse scale involves the most smoothing, there remains some ambiguity about the precise location of the matching sub-contour. Just as archaeologists do by hand, we leverage finer-scale detail to validate the match. Using one region as a template, we refine the location of the matching region via normalized cross-correlation image registration. In Figure 3.4b we show the result of this step which resembles the "lock-in" effect archaeologists use to describe the certainty that comes from aligning small scale features. Note that the matching region of the smaller piece has moved right in the scale-space representation and clockwise around the piece.

c) Fan-out estimation The first and last points of every correctly matching contour sit next to a non-matching point. In Figure 3.4, for example, the match stops when the two contours turn sharply away from each other. This results in contrasting values in the two contour description strings encoded at the base of the scale-space representations. As we increase the scale, and move up the scale-space representations, these non-matching values are smoothed with an increasing number of matching values (as the standard deviation of the smoothing function σ_r increases). The result of this smoothing is an iterative decreasing in the number of matching values. To address this effect, which we call *fan-out*, we incrementally expand the base of the matching regions until a threshold is surpassed. As shown in Figure 3.4c, the resulting trapezoidal regions correctly convey the full match.

3.5 Pairwise alignment

Once two matching boundary regions have been identified, the second step of our algorithm is to estimate the transformation that brings the fragments into alignment. We formulate this as a least-squares optimization problem where the error is measured at each sampled point of the matching boundary contours according to two metrics. The first, which we call "contour error," quantifies the distance between the corresponding points. This ensures that the fragments are tightly aligned. The second, which we call "surface error," quantifies the geometric continuity across the points. This ensures that the resulting surface geometry transitions smoothly from one fragment to the other.

More precisely, given two fragments $\mathbf{A}$ and $\mathbf{B}$, we seek the transformation $T_{\mathbf{B},\mathbf{A}}$ that brings $\mathbf{B}$ into alignment with $\mathbf{A}$ such that the sum of squared residuals $\sum_i^m e_i^2$ across the m points of their matching boundary contours is minimized. We formulate the residual at point i as a weighted combination of our two metrics,

$$e_i = e_i^{\mathrm{c}} + \alpha e_i^{\mathrm{s}} \ , \tag{3.6}$$

where e_i^{c} and e_i^{s} are the contour error and surface error, respectively, and α is a relative weight.

Contour error

To formulate the contour error e^{c}, we let $\{\boldsymbol{a}_i \mid 1 \leq i \leq m\}$ be the m 3D contour points from $\mathbf{A}$, and let $\{\boldsymbol{b}_i \mid 1 \leq i \leq m\}$ be the corresponding points from $\mathbf{B}$. The contour error e_i^{c} at point i is then the Euclidean distance between corresponding points,

$$e_i^{\mathrm{c}} = \| T_{\mathbf{B},\mathbf{A}}(\boldsymbol{b}_i) - \boldsymbol{a}_i \| \ . \tag{3.7}$$

The transformation $T_{\mathbf{B},\mathbf{A}}$ is defined in terms of a rotation matrix $\mathbf{R}$ and a translation vector $\mathbf{t}$,

$$T_{\mathbf{B},\mathbf{A}}(\boldsymbol{b}_i) = \boldsymbol{b}_i \mathbf{R} + \mathbf{t} \ . \tag{3.8}$$

This error metric is insufficient on its own because the contours do not convey any reliable information about the fragment surfaces themselves; the alignment should not only be tight, but should result in a smooth surface.

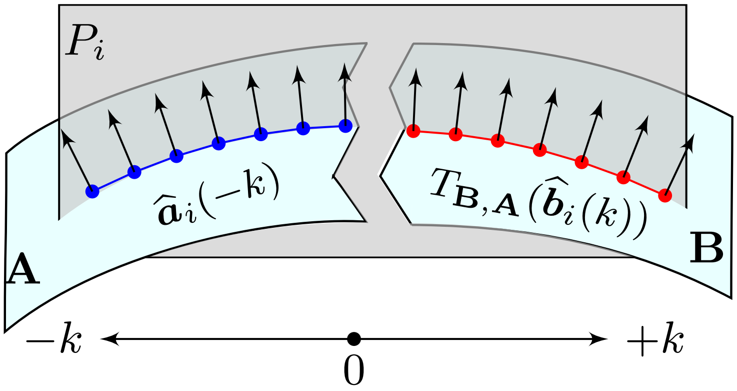

P_i
$\widehat{\boldsymbol{a}}_i(-k)$
$T_{\mathbf{B},\mathbf{A}}(\widehat{\boldsymbol{b}}_i(k))$
A
B
$-k$
$+k$
0

Figure 3.5: Pairwise alignment. When aligning fragment ***B*** to fragment ***A*** we seek to optimize the surface continuity across the fragments. To evaluate this resulting continuity at point i, we intersect the two surfaces with a plane P_i perpendicular to the boundary contour. We then extract surface normals at regular intervals from both fragments. The transformation $T_{\mathbf{B},\mathbf{A}}$ is then evaluated in terms of the gradient of these normals.

Surface error

Our second error term evaluates the alignment quality of the fragment surfaces. adjoining surfaces. We evaluate the continuity of the newly-formed surface in terms of the progression of surface-normal vectors across the matching boundary contour points. In Figure 3.5 we illustrate how the surface error e_i^{s} is computed for a point i. We begin by computing the plane P_i that is orthogonal to the boundary contour's tangent vector at this point. The intersection of this plane with both surfaces forms a curve (shown in red and blue). Along this curve, we extract surface normals from the fragment surfaces at regular intervals. Specifically, we let $\widehat{\boldsymbol{a}}_i(j)$ denote the j^{th} surface normal from point $\boldsymbol{a}_i$, and define $\widehat{\boldsymbol{b}}_i(j)$

analogously. We then form a piecewise function $\boldsymbol{q}_i(k)$ of these normal vectors as

$$\boldsymbol{q}_i(k) = \begin{cases} \widehat{\boldsymbol{a}}_i(-k) & \text{for } k < 0 \\ \mathbf{0} & \text{for } k = 0 \\ T_{\mathbf{B},\mathbf{A}}(\widehat{\boldsymbol{b}}_i(k)) & \text{for } k > 0 \ , \end{cases} \tag{3.9}$$

where $T_{\mathbf{B},\mathbf{A}}$ is the alignment transformation to be optimized. Note that we negate k in the first case since $\widehat{\boldsymbol{a}}(\cdot)$ is only defined for positive inputs.

An optimal alignment will result in a gradient of surface normals $\frac{\partial \boldsymbol{q}_i}{\partial k}$ whose value at the boundary (i.e., $k = 0$) matches the gradient of surface normals on the adjoining fragments. We therefore compute a target gradient value equal to the mean of the gradients from both fragments evaluated a short distance ϵ from the boundary contour. The surface alignment residual error is then computed as

$$e_i^{\mathrm{s}} = \frac{\partial \boldsymbol{q}_i}{\partial k}(0) - \frac{1}{2}\left[\frac{\partial \widehat{\boldsymbol{a}}_i}{\partial j}(\epsilon) + \frac{\partial \widehat{\boldsymbol{b}}_i}{\partial j}(\epsilon)\right] \ . \tag{3.10}$$

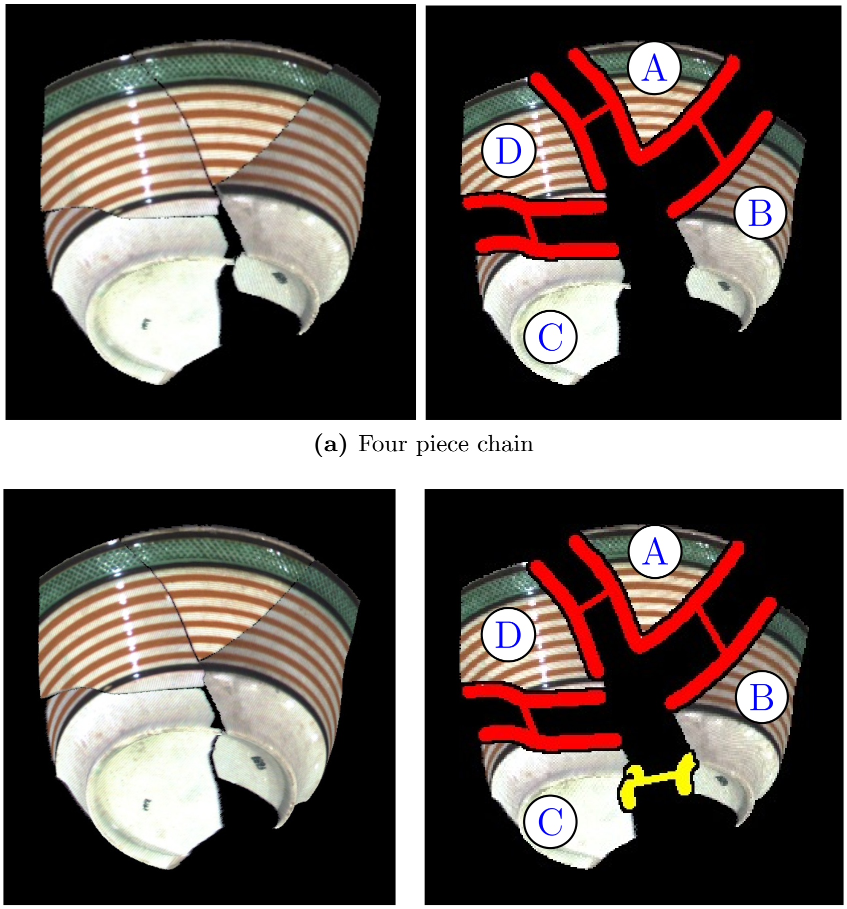
To compute the derivative of $\boldsymbol{q}$ at point i, we extract fifteen normals $\{\boldsymbol{q}_i(k) \mid -7 \leq k \leq 7\}$ from a thin strip of the corresponding surfaces. By restricting this calculation to a thin strip we avoid any assumptions about the global geometry of the object. We then weight these 15 normals using a standard discrete derivative kernel. For the target gradients $\partial \widehat{\boldsymbol{a}}_i$ and $\partial \widehat{\boldsymbol{b}}_i$, we use the same process and evaluate the gradient at the midway point $\epsilon = 4$. We then solve this system with Levenberg-Marquardt iterative minimization [53, 54].

To provide a measure of similarity between the two error terms e^{c} and e^{s}, we set the relative weighting term α (from Equation 3.6) equal to the boundary contours' sub-sampling distance divided by π (the maximum value for e^{c}). To reduce the running time of this optimization, we pre-compute the intersection planes, surface normals, and target gradients for each contour point. By doing so, however, we are making the assumption that the intersection plane computed for fragment $\boldsymbol{A}$ at point i is parallel to the intersection plane computed for fragment $\boldsymbol{B}$. This assumption may be violated, in particular when alignments are re-optimized, as we discuss next. To address this, we add an additional error term that quantifies the orientation error of these planes. Since the range of this error is the same as for e^{s} we also weight it with α.

3.6 Incorporating context

As illustrated in Figure 3.6, the final component of our framework is to incorporate the global geometry of the object as it becomes known. We use each additional match to increase the overall alignment quality of the object, which in turn improves the quality of future matches. We encode the global geometry as a graph in which the nodes represent individual fragments, and the edges correspond to matching contours between them. Cycles in this graph correspond to subsets of fragments whose alignments depend on each other, i.e. changing the orientation of one fragment in such a subset will impact the alignment quality of the others.

Figure 3.6 shows the significance of a cycle in the graph. In the four node chain of Figure 3.6a there is a gap between fragments $\boldsymbol{B}$ and $\boldsymbol{C}$ that has resulted from the accumulation of small alignment errors in the three edges. In Figure 3.6b, the addition of the edge connecting fragments $\boldsymbol{B}$ and $\boldsymbol{C}$ has created a cycle. It is plain to see that aligning the two fragments using only this newly found matching contour (marked in yellow) will decrease the quality of the neighboring alignments. It would also be inaccurate to simply move fragment $\boldsymbol{B}$ evenly between its neighbors. Instead, this additional edge provides important context information that should be used to improve all of the other alignments.



(a) Four piece chain

(b) Four piece cluster

Figure 3.6: Incorporating context. The configuration of fragments as they are incrementally matched to each other forms a graph structure. As matching contours are found, edges are added to the assembly graph. When cycles are detected, as in (b), we jointly re-optimized the constituent alignments to incorporate the additional contextual information. Note how the gap between pieces ***B*** and ***C*** has been closed with the addition of the fourth (yellow) edge.

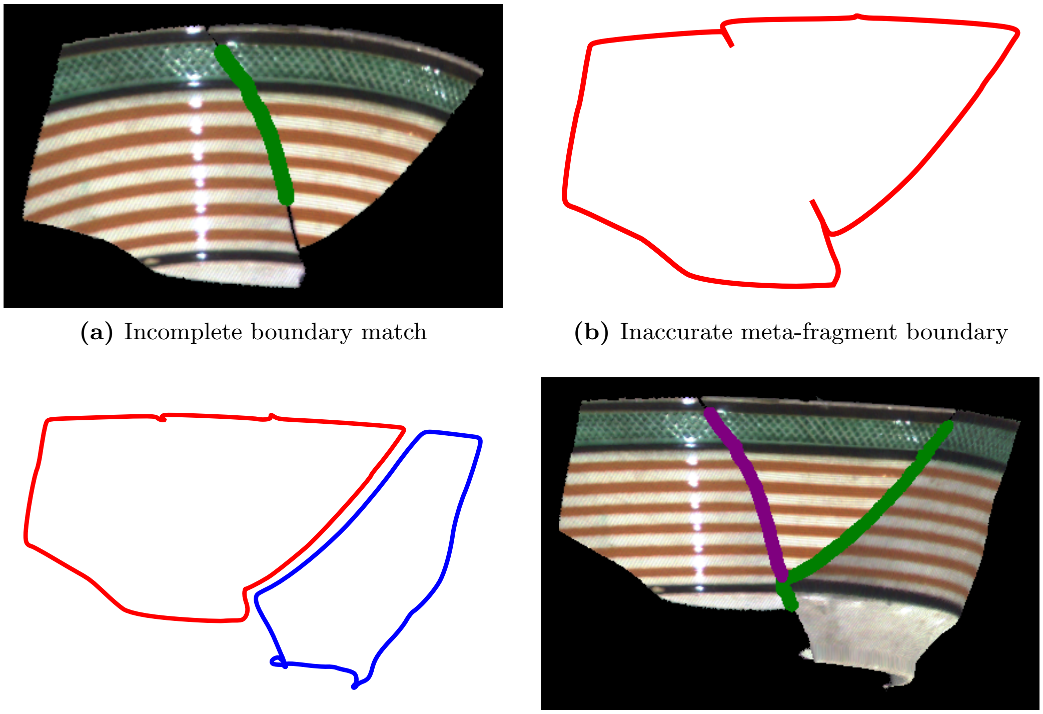
We detect cycles by computing all of the paths between the fragments involved in the new match. This is done simply with a single depth first traversal of the graph. Using the correspondences embedded in every edge of the cycle to evaluate the overall error in the subset, we jointly re-optimize the constituent alignments. Note how the gap between ***B*** and ***C*** has been closed by this process, and the overall alignment quality of the subset has been improved.

We formulate the update alignment of the subset as a least squares optimization where the residual error is measured across all points of every matched contour pair according to Equation 3.6. All alignment translations and rotations are optimized jointly to minimize the error of the entire subset. As the number of cyclically dependent nodes approaches the entire set of fragments, this re-optimization becomes a full global optimization. By building up the global model in a bottom-up fashion, however, each re-optimization is simply a refinement of the previous configuration. Thus, we leverage the accurate estimates of our pairwise alignments to reduce the search-space of the global optimization.

After updating the alignment of the subset, the set of points being used to evaluate an alignment may no longer be an optimal set of correspondences. Recall that the alignment quality across a pair of matching contours depends on the distance between pairs of corresponding points. When a cycle is formed, we may find that one fragment should slide along the other to improve the overall alignment quality. To that end we utilize the core functionality of Besl and McKay's iterative closest point (ICP) algorithm [55]. Specifically, we iterate between optimizing the transformations of the subset, and updating the correspondences used to evaluate them. At each iteration every point is paired with the nearest point on the corresponding contour. Once the correspondences become stable, the optimal alignment for the object will be found.

3.7 Grouping fragments

The final aspect of our method is designed to aid in the detection of matching contour groups such as the one shown in Figure 3.7. When two or more pieces are bound together by matching sub-contours, they effectively form a single *meta-piece*. Intuitively, this corresponds to virtually gluing pieces together–the matched sub-contours become internal to the meta-fragment. Each connected component subgraph of our alignment graph is given



(a) Incomplete boundary match

(b) Inaccurate meta-fragment boundary

(c) Corrected meta-fragment boundary, and next fragment

(d) Multi-fragment match found

Figure 3.7: Grouping fragments. Sets of connected fragments are treated as a single "meta-fragment" and given their own boundary-contour representation. Incomplete matches (a) will result in inaccurate meta-fragment boundaries (b). To account for this we first remove overlapping regions. We then smooth the junction points so that the geometry will mirror that of the missing piece (c). In this case, the next match found spans both fragments of the meta-fragment (d).

its own scale-space boundary representation which is then compared against the remaining unpaired fragments and meta-fragments.

The boundary representation for a meta-fragment is formed by first extracting the unmatched boundary points from all constituent pieces. These points are concatenated to from a single boundary contour. As shown in Figure 3.7, there are two important factors to consider during this process. First, the matching boundary contour between the two fragments may have been too short. This will result in small sub-contours that essentially overlap each other as in Figure 3.7b. To account for this, we extend the boundary contours as shown in Figure 3.7c. The second factor to consider comes from our preprocessing step. Recall that each fragment's raw boundary contour is initially smoothed and sub-sampled, causing a slight rounding of sharp corners. As shown in Figure 3.7c, the meeting point of two fragments must be smoothed to account for this.

This approach ensures that contour regions that have previously been matched are no longer considered for future pairings. Additionally, the boundary contour of a meta-fragment will typically have a more favorable representation. In Figure 3.7d, the third fragment is successfully paired with both of the other fragments. Since we prioritize candidate matches based on their length, the small matching contour that makes up the bottom of this match would not otherwise have been found in a reasonable time-frame. Further examples can be seen in the final steps of Figure 3.8, and in the begging of the third row of Figure 3.12.

3.8 Experimental evaluation

To validate our algorithm we augmented it with an simple, interactive user interface. At each iteration of our algorithm (summarized in Figure 3.1) we first extract the longest matching boundary contour. We then align the constituent fragments and present the updated reassembly to the user who may then accept or reject it. When a match is accepted, it is added to the reassembly, and the additional context information is propagated to the other alignments. Except where indicated, the resulting reassemblies contain every fragment in the dataset. Each dataset is incomplete, describing only a partial vessel. Despite this, we are able to successfully recover the full 3D geometry of every dataset. Unfortunately, there is no ground truth to which we could compare the exact alignment of the fragments.

In Figure 3.8 we show the partial reassembly of a store bought vase. At each step, the current candidate match is shown in green, other matches that have been re-optimized

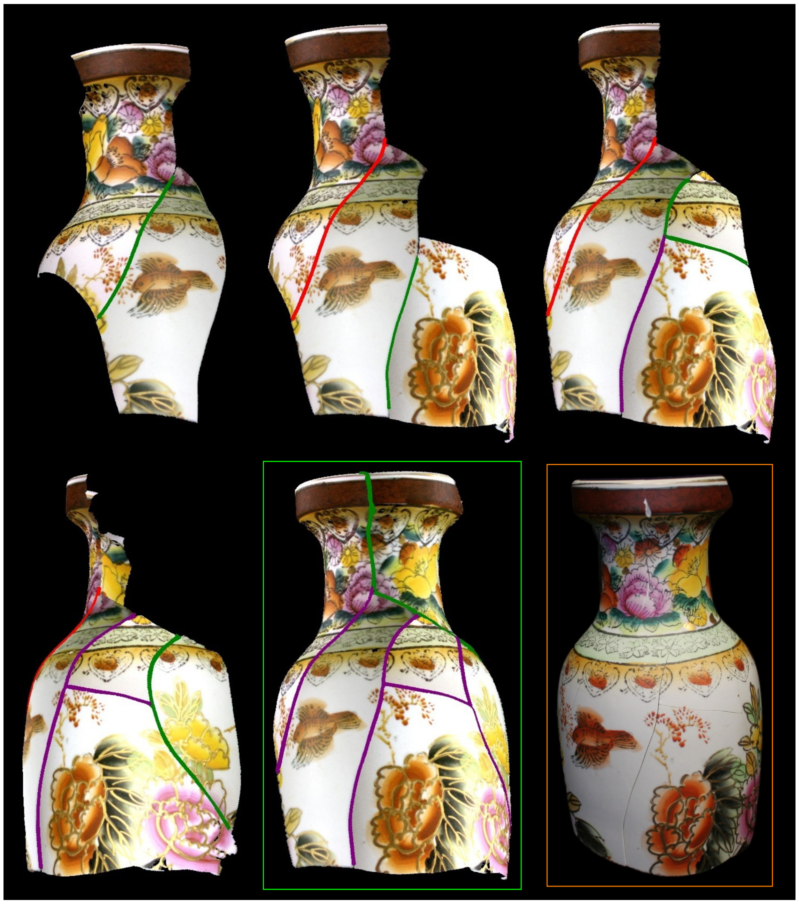


Figure 3.8: 6 piece vessel reassembly. 6 fragments from a store bought vase are reassembled using our system (green box) and by hand (orange box). At each step the green contour indicates the proposed addition, and purple indicates alignments that have consequently been adjusted. Note how when the last fragment is added all of the other alignments have turned purple to indicate that additional contextual information has been used to improve them. This dataset is labeled Vase in Figure 3.9.

are colored in purple, and any other past matches are shown in red. In this artificial example, each of the suggested matches is correct making the full reassembly essentially fully automatic. As such, the full process requires only about 12 seconds.

In Figure 3.9 we summarize our results and compare total reassembly times using our method with manual reassembly of the actual artifacts. In this graph, the store bought vase of Figure 3.8 referred to as Vase. The remaining five objects are recently excavated cultural heritage artifacts, and as such exhibit increased chipping, signs of wear, and a greater range

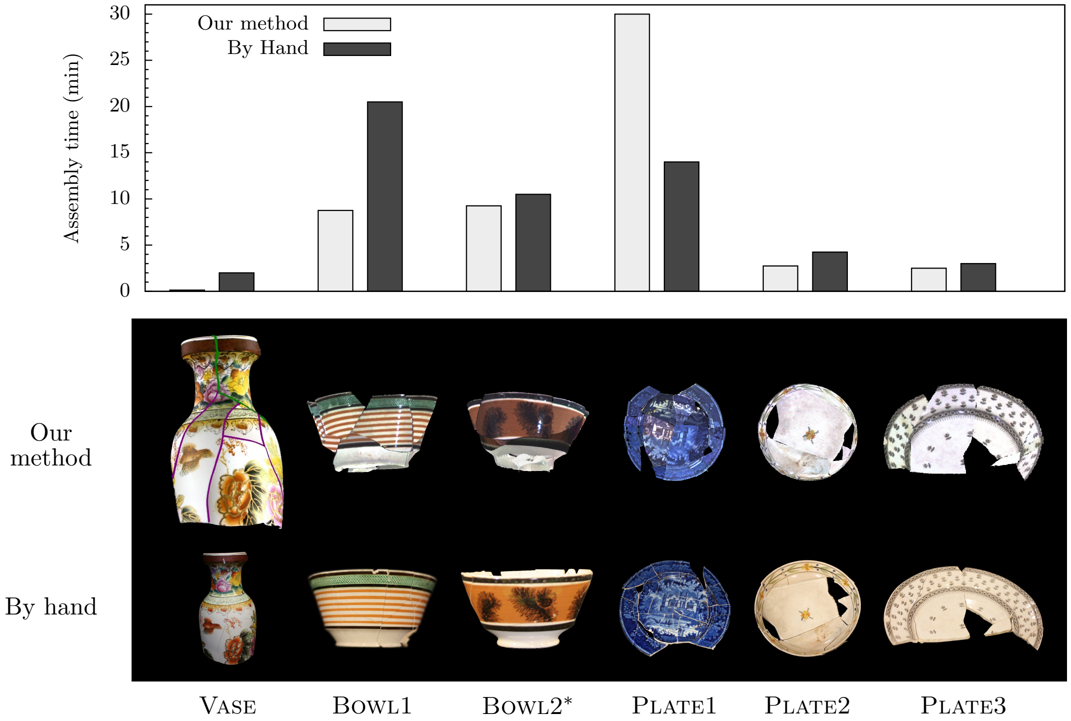


Figure 3.9: Reassembly times in minutes. Our method (light) enables more rapid reassembly than by hand (dark). Note that when reassembling Bowl2 by hand, our test subject was unable to complete the task and gave up at the reported time. The long reassembly time of Plate1 (shown in Figure 3.12) is due to the strongly connected nature of the reassembly graph. Large scale re-optimizations are far more frequent when assembling this artifact.

of fragment size. Although these datasets describe axially symmetric objects, the fragments themselves are quite small. Consequently, past methods that rely on estimating the profile contour of symmetric objects [13, 14] are not be well suited for this task.

In Figure 3.10 we show the steps taken by a user to reassemble 15 pieces of one such historical artifact. Note, however, that we have omitted a small triangular piece in the top of the rim. This omission is visible in the final reassembly (shown in a green box) as compared to the ground truth image (shown in an orange box). This piece was too small to be reliably analyzed, due to the fan-out effect discussed in Section 3.4. This example is referred to as Bowl1 in Figure 3.9.

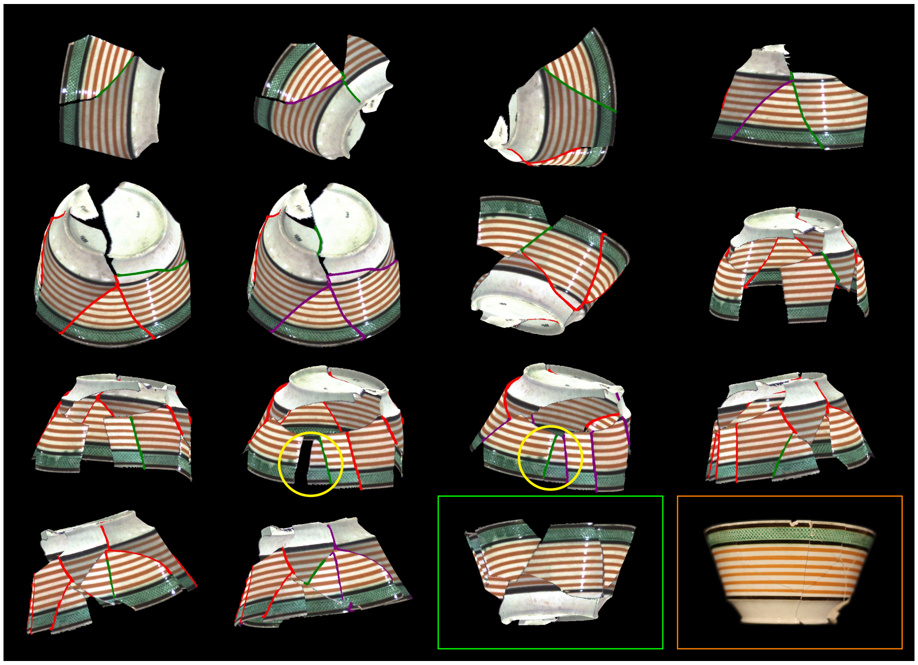


Figure 3.10: 16 piece vessel reassembly. A 16 piece vessel is reassembled using our system (green box) and by hand (orange box). The green contours indicate our method's proposed addition, purple contours indicate cyclically dependent alignments that have been re-optimized to include the additional context information provided by the green contour. Circled in yellow, this re-optimization is best illustrated in the transition between the middle two images of the third row where a large gap is closed. This result is labeled Bowl1 in the graph of Figure 3.9.

In Figure 3.11 we show the reassembly of an 11 piece historical artifact. Note how a large gap is closed in the last step, as indicated by the yellow circles. Since many pieces are missing on this side of the bowl, a long chain of pieces has accumulated error. When the matching contour is found that closes this gap, the entire rim of the vessel is re-aligned to distribute the error. This example is referred to as Bowl2 in Figure 3.9.

Figure 3.12 shows our most challenging dataset. Whereas the painted scene provides useful clues to archaeologists working with the artifact itself, the color information contained at the boundary of each piece is too subtle to be discerning in our scale-space images. Further, there are many small missing fragments that partition long matching contours into smaller matches. Because of these two factors many erroneous matches are considered. We

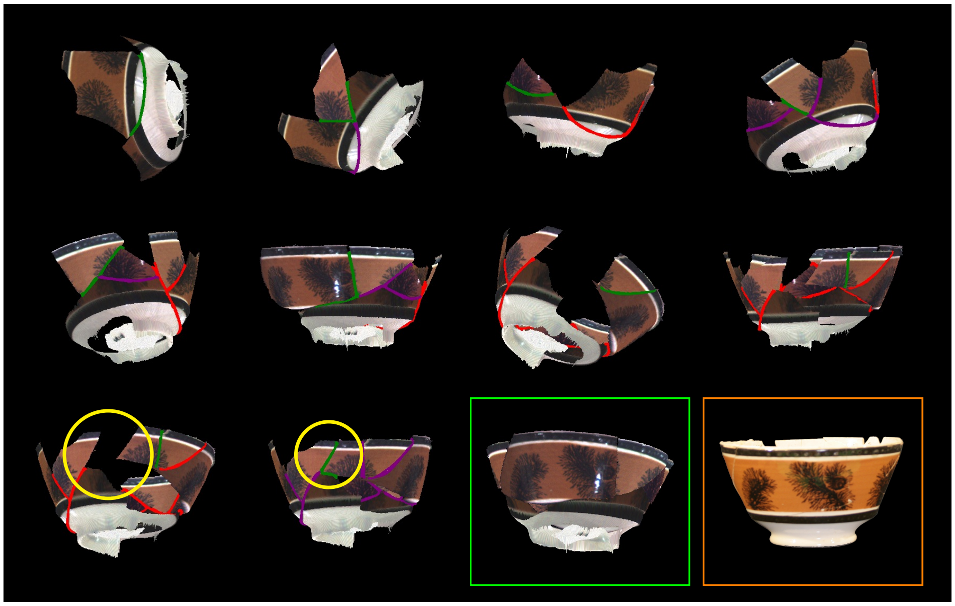


Figure 3.11: 10 piece vessel reassembly. A 10 piece vessel is reassembled using our system (green box) and by hand (orange box). The entire process takes only 9 minutes with our system compared to 20 minutes by hand. Note the large (circled) gap that is closed in the last step. Shown in purple, nearly every alignment is adjusted to correct this accumulated error. This dataset is labeled Bowl2 in the graph of Figure 3.9.

discard clearly erroneous matches by inspecting the resulting alignment error. If too much space exists between matching contours, or the resulting surface is not smooth, the match is discarded. This dramatically reduces the amount of user interaction required in all cases. In this case, however, evaluating alignment error may involve large cycles of fragments. Note that almost every step involves a large scale re-optimization. As shown in Figure 3.9, this vessel (referred to as Plate1) is the only example where assembly by hand took less time than using our method. Nevertheless, despite the challenges in this dataset, our method is able to rapidly align all but one fragment of this object.

Figures 3.13 and 3.14, show the final two datasets, which are respectively labeled Plate2 and Plate3 in Figure 3.9. These examples show the importance of our surface error alignment metric. Many of the matching contours are essentially straight lines. As such, simply minimizing the distance between the contours would result in dramatic surface discontinuity across the matching regions. By additionally optimizing the geometric continuity across the surfaces, the pieces are accurately aligned.

Figure 3.9 shows the reassembly times for our six datasets using our method (light) and by hand (dark). Although our user interface is rudimentary, it is important to note that with the exception of Plate1 our system allows for more rapid reassembly than by hand. Bowl2, which is detailed in Figure 3.11, proved too challenging to reassemble by hand; the time reported is when the test subject abandoned the task. We believe this is due to the object's many similarly shaped fragments. By utilizing the full scale-space of each fragment, we exploit subtle detail to eliminate the guesswork associated with manual reassembly.

3.9 Summary

In this chapter we presented a method for reassembly that makes no assumptions about the global geometric or photometric structure of the object. It is therefore applicable to any reassembly problem where the boundary of each fragment may be extracted. This includes volumetric object reassembly, with the added benefit that a full 3D model of each fragment is not needed. To achieve this, we have presented a method that balances our novel pairwise alignment method with a flexible global approach that leverages the important context information each match provides. By allowing the global model to be uncovered incrementally, we successfully complete full reassemblies even when many of the object's fragments are missing. Since access to cultural heritage objects is greatly restricted, achieving reassembly of real artifacts in a virtual setting is itself an important contribution.

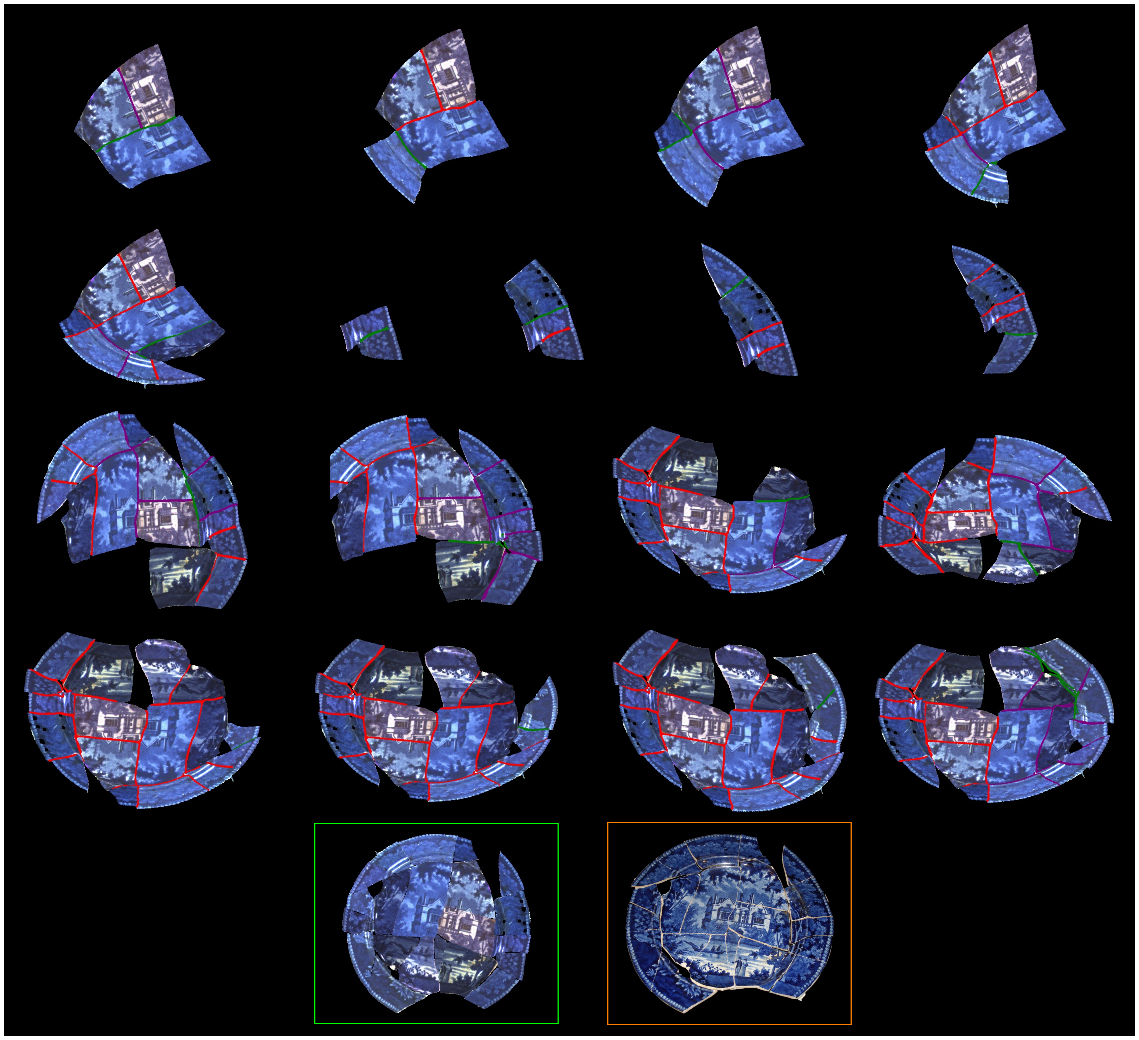


Figure 3.12: Steps taken to reassemble a 21 piece artifact. Note how the primary cluster is set aside after the eighth piece. At this point the strongest matches were part of the rim. After completing this 4 piece section, a connection was made between the two components, enabling the completion of the rest of the plate. The final result (green) is compared to the hand reassembly (orange). Note that a few steps have been omitted for compactness. This vessel is referred to as Plate1 in the graph of Figure 3.9.

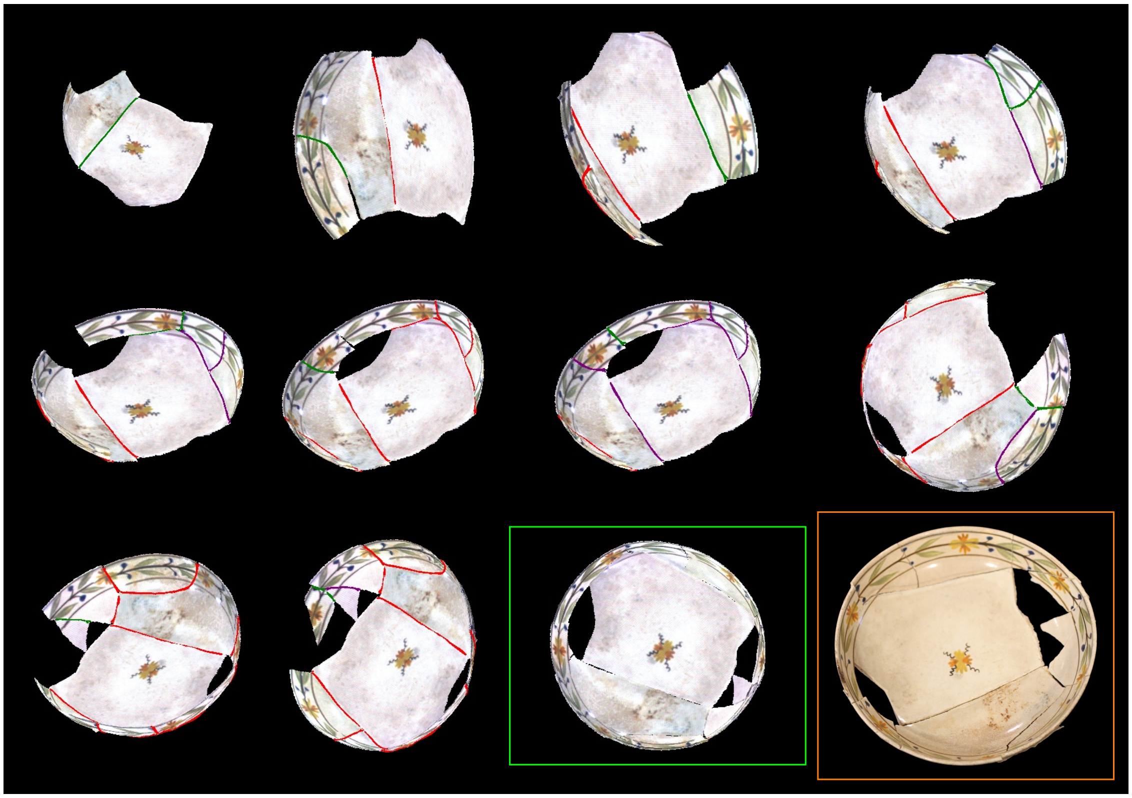


Figure 3.13: This 11 piece vessel is labeled PLATE2 in Figure 3.9.

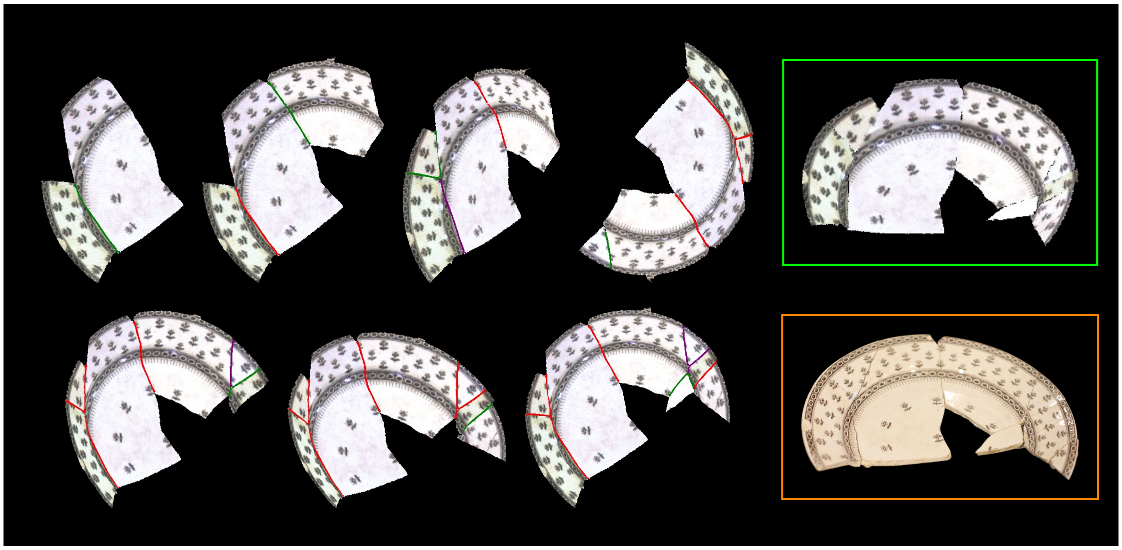


Figure 3.14: This 8 piece vessel is labeled PLATE3 in Figure 3.9.

Chapter 4: Geometric texture analysis

4.1 Introduction

When we look at the world around us, we don't just see patterns of color, we see texture. Thick tree bark, for example is more than jut a mix of brown hues. After all, if the bark is painted white, we could still identify it as bark. Even with eyes closed we can touch and immediately identify bark, toast, sand-paper, etc. In other words, texture is a *geometric* phenomenon.

Despite this, past work on texture analysis has focused on the 2D appearance of texture. Whereas the appearance of a texture depends greatly on illumination and viewing conditions, the underlying 3D geometry remains constant. In this chapter we pursue a deeper understanding of texture geometry in order to enable important tasks, such as object recognition and scene segmentation, to be done reliably in any condition. To do so, we study the two primary characteristics of geometric texture—self similarity and scale variability—and exploit them to perform geometry recovery and geometric texture recognition.

Textures, whether they are regular or stochastic, exhibit some form of repetition in their underlying geometry. Our first contribution is a photometric stereo method designed to extract this structure from images of the surface, and use it to recover the geometry of the texture. We observe that the appearance of each point is determined fundamentally by both the surface orientation at that point, and whether or not it is being directly illuminated. Since each of these factors cannot be estimated without knowledge of the other, we must estimate them jointly. To do so, we derive a probabilistic graphical formulation in which these two conditionally independent factors are linked by their combined effect on the appearance of each point. We sequentially order the dense set of observations as the light is waved over the texture. This allows us to derive novel priors for both factors. In particular, we leverage the similarity of geometry that is present in spatially disjoint locations as a prior on the surface normals. Pixels whose intensities vary similarly as the light moves must have similar orientations. This increased contextual information allows us to minimize the effect of self-occlusion and interreflection.

Our second main contribution is a rotation-invariant scale-space representation of the geometry of texture. Our aim is to compactly encode how the apparent texture geometry varies with its distance from the observer so that a texture can be identified reliably regardless of its distance and orientation. Since the 3D surface geometry itself cannot be reliably recovered due to discontinuities of the surface, we directly use the surface normal field recovered with the proposed texture photometric stereo as the foundation for our representation. We first represent the surface normals using their spherical coordinates. This allows us to describe the geometry using a 2D histogram in which rotations of the texture correspond to shifts along the azimuthal axis. By applying the 2D Fourier transform we obtain a rotationally invariant representation of the texture geometry. Next, we compute the scale space of the texture geometry by convolving the surface normal field with Gaussian kernels of increasing standard deviations until the surface normal field is completely coherent. We show that this scale space of geometric texture accurately captures the evolution of the apparent surface normal field over increasing distance, and as such, approximates the underlying geometric scale space.

Our final contribution is a new texture database consisting of over $40,000$ images that we use to demonstrate the descriptiveness of our scale-space representation. We perform extensive classification experimentation with query textures at multiple distances, with varying degrees of in-plane and out-of-plane rotations as well as under simulated noise. In total, we test our representation on 600 combinations of distance and orientation. The high accuracy of the classification results indicate the descriptive yet compact nature of our texture representation, and demonstrates the importance of geometric texture analysis.

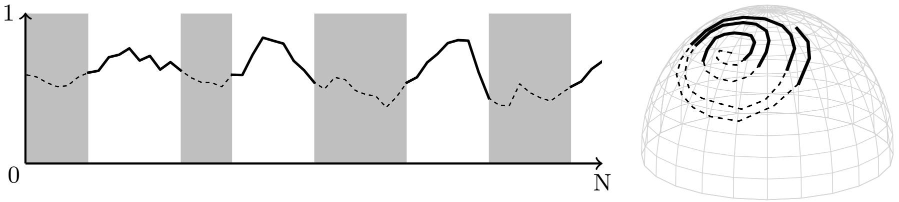
4.2 Related work

Past work on texture analysis has focused primarily on the 2D imaged appearance of textures. In this domain, many methods have been proposed for rotation invariant and scale invariant classification [17, 18]. Similarly, a number of methods have been proposed for scale-space analysis of texture appearance [56]. The appearance of texture, however, depends fundamentally on the underlying surface geometry. Unlike its appearance, the geometry of a texture does not change with the illumination and viewing conditions. In this work we focus directly on the geometry of surface texture and derive a rotationally invariant scale-space representation which we use to perform classification.

There has been some attention to illumination invariance. Smith [57] use photometric stereo to recover surface normals and then encode them as a surface orientation histogram. This approach, however, is geared at analysis to find surface defects, whereas our method is being used to perform classification. McGunnigle and Chantler [58] and Barsky and Petrou [59] extract and use explicit 3D shape and surface reflectance information to perform classification. Penirschke et al. [60] propose an illumination invariant classification scheme based on earlier work by Chantler et al. [61] wherein they study the effect of changes in illumination direction on the image texture. Other approaches have involved explicitly training a classifier on multiple images of the texture from different viewpoints and illumination conditions [19, 20]. Dana and Nayar [62] model texture surface as a Gaussian-distributed random height field. To avoid this assumption we develop a novel scale-based representation in the frequency domain. Although some authors have considered the effect of the underlying geometry, by studying it directly we deepen the understanding of texture, and enable improvements to appearance modeling and synthesis.

In order to analyze the scale variability of geometric texture, dense and accurate surface normal information must be gathered. As first introduced by Woodham [21], the technique of photometric stereo enables the extraction of accurate surface orientations for simple Lambertian objects. As noted by Nayar et al. [22], problems arise when self-occlusions cause shadows, or interreflection exaggerates illumination conditions. Although there have been many advances in photometric stereo to these, and other ends, our method tackles the less-studied problem of estimating the geometry of textures.

To address self-occlusions, Barsky and Petrou [63] assume that each pixel is directly illuminated in at least two images. They then ignore light source directions where the observed intensity is low. Chandraker et al. [64] model light source visibility using a Markov Random Field with a strong spatial prior that encourages detection of large shadows. Similarly, Sunkavalli et al. [65], cluster pixels into "visibility subspaces" by observing the effect of large shadows. Nayar et al. [66] project a moving pattern onto the scene to separate the effect of direct illumination from that of interreflection and other "global" illumination phenomena. Wu and Tang [67] collect a dense set of images and use their inherent redundancy to detect and discard the observations that are most impacted by global illumination. We further exploit redundancy by leveraging the repetitious nature of texture. In a manner similar to that of Koppal and Narasimhan [68], we manually wave a light source over the texture and detect similarly oriented surface patches in physically separate locations based on the similarity of their appearance variation. Unlike Koppal and Narasimhan, we use the full appearance profile, not just extrema. This allows us to establish a non-parametric



(a) Observed intensities

(b) Lighting directions

Figure 4.1: Observed intensity profile and corresponding relative lighting direction at a single pixel. We use the intensity profile of a pixel (a) to estimate its light-source visibility. Shown as thick curves, convex regions of the (smoothed) intensity profile serve as good initial estimates of when a pixel is directly illuminated as they correspond to times when the light source passes near the surface normal (b).

distribution of orientation vectors that we use as a prior in our expectation maximization framework.

4.3 Texture photometric stereo

Textures, whether they are regular or stochastic, exhibit some form of repetition in their underlying geometry. We exploit this repetition in our texture photometric stereo in order to recover accurate surface normals.

4.3.1 Robust estimation

As light passes over a Lambertian texture the intensity I_t^x of a pixel x rises and falls depending on the relative angle of the light source direction $\boldsymbol{l}_t$ at that time t, and the surface orientation $\boldsymbol{n}^x$ at that point according to the basic Lambertian reflectance model

$$I_t^x = \rho^x \max(0, \boldsymbol{n}^x \cdot \boldsymbol{l}_t) \ , \tag{4.1}$$

where ρ^x is the albedo value at that point. Following Koppal and Narasimhan [68], as the light is waved over the texture, we take k discrete observations, and call $\boldsymbol{I}^x = \{I_t^x \mid t =$

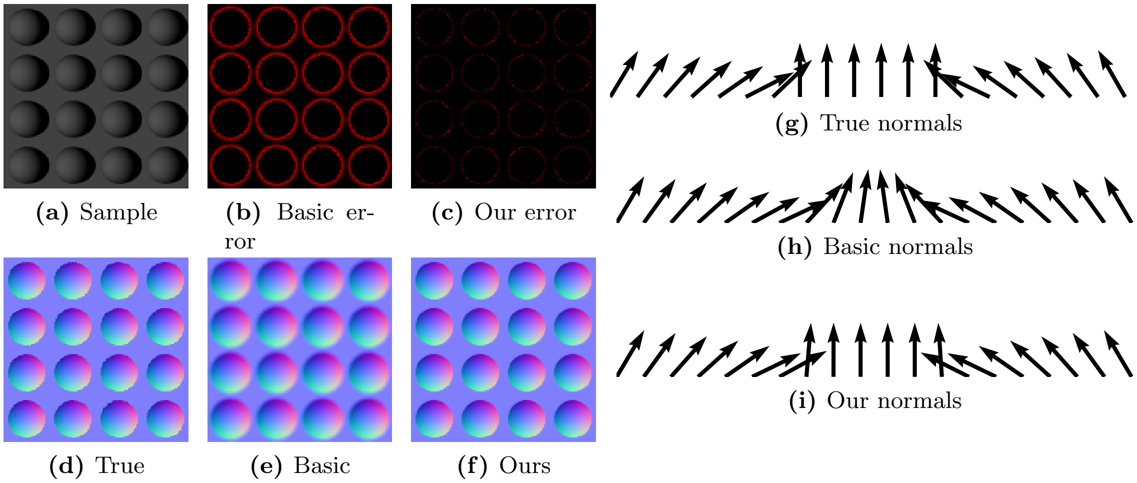


Figure 4.2: Synthetic scene ground truth comparison. Without any model of light-source visibility, shadows smooth the recovered normal field (e). Using our method results in an accurate estimation (f). Brighter pixels in the error fields of (b) and (c) correspond to greater angular error. In the right half we show a slice of normals between two spheres. Note how the normals between the spheres in the basic method results (h) have been corrupted. Ray tracing interpolation is responsible for slight inconsistencies between our results and the ground truth.

$1, \ldots, k\}$ the *intensity profile* of pixel x. Figure 4.1 shows the path of the moving light source on the unit hemisphere centered at a surface point. If we align the north pole of the hemisphere with the surface orientation at that point, then in accordance with Equation 4.1, as the light moves closer to the pole, the observed intensity will increase.

Unfortunately, this model ignores a fundamental factor in appearance—light source visibility. Small scale self-occlusions lead to shadows in small regions and interreflection will increase the observed intensity. As shown in Figure 4.2, failure to disregard these observations will result in inaccurate estimations of the underlying normal field. In our synthetic experimentation, our method results in a median error of 2.3°, while the traditional method results in a median error of 14.8°.

We use a binary visibility function V_t^x to indicate whether or not pixel x at time t is directly illuminated,

$$V_t^x = \begin{cases} 1 & \text{if pixel } x \text{ is directly illuminated} \\ 0 & \text{otherwise .} \end{cases} \tag{4.2}$$

By applying this function to both the observed intensity I_t^x, as well as the predicted intensity, we are able to effectively ignore misleading values. This augmented version of the Lambertian reflectance model may be written as

$$V_t^x I_t^x = V_t^x \rho^x \boldsymbol{n}^x \cdot \boldsymbol{l}_t \ , \tag{4.3}$$

where the max function from Equation 4.1 has been absorbed into V. Since the surface normals are unit vectors, we simplify this notation by representing the albedo-scaled normal as $\boldsymbol{N}^x = \rho^x \boldsymbol{n}^x$, where $\rho^x = \|\boldsymbol{N}^x\|$.

A simple approach to estimating $\boldsymbol{V}$ would be to set $\boldsymbol{V}_t^x = 0$ for low intensity observations. For pixels with low albedo values, however, this is too limiting. A better approach is to use the portions of the light trajectory that are near the normal direction. As shown in Figure 4.1a, we estimate these regions as convex portions of the intensity profile. To account for noise, we first smooth the intensity profile with a Gaussian distribution of standard deviation 3 before taking the second derivative with respect to t

$$\boldsymbol{V}_t^x = \begin{cases} 1 & \text{if } (\mathcal{N}(3) * I^x)''(t) < 0 \\ 0 & \text{otherwise ,} \end{cases} \tag{4.4}$$

where $*$ denotes convolution. These regions, which are shown as thick solid lines in Figure

4.1a, correspond well with lighting directions near the surface orientation (b).

4.3.2 Joint visibility and normal estimation

Note that even under direct illumination, the effects of interreflection, subsurface scattering and other global illumination phenomena are impossible to avoid. Our goal is to utilize the observations where the appearance is dominated by the direct illumination component which will contain a more reliable encoding of the surface normal. With this in mind, estimating the visibility function is more akin to robust estimation wherein we seek to ignore as many outliers as possible while simultaneously avoiding conclusions based on sparse observations. Assuming known light source directions, estimating the optimal visibility function will then result in accurate surface normal estimations. In order to recover the visibility function, however, we must already know the correct surface orientations or we may be misled by low albedo values. In other words, since both the visibility function, and the surface normals contribute to the appearance of the texture, they must be estimated jointly.

To refine the visibility $\boldsymbol{V}$ and normal field $\boldsymbol{N}$ estimates we propose a probabilistic graphical model similar to a Factorial Markov Random Field [69]. We represent the visibility function and the surface normals as two separate, conditionally independent latent layers that are linked by their contribution to the observed appearance layer $\boldsymbol{I}$. Using this formulation, we estimate the surface orientations and visibility functions jointly, by maximizing the posterior probability

$$p(\boldsymbol{N}, \boldsymbol{V} \mid \boldsymbol{I}) \propto p(\boldsymbol{I} \mid \boldsymbol{N}, \boldsymbol{V}) p(\boldsymbol{N}) p(\boldsymbol{V}) \ , \tag{4.5}$$

where $p(\boldsymbol{I}|\boldsymbol{N}, \boldsymbol{V})$ represents the likelihood of the observations given the estimated normals and visibility function, and the priors $p(\boldsymbol{N})$ and $p(\boldsymbol{V})$ encourage solutions that characterize the repetitious nature of the texture.

Likelihood

We assume that the noise inherent in the observations is normally distributed, with a common variance σ^2

$$p(\boldsymbol{I}|\boldsymbol{N},\boldsymbol{V}) \propto \prod_t \prod_x \mathcal{N}\left(V_t^x I_t^x - V_t^x \boldsymbol{N}^x \cdot \boldsymbol{l}_t, \sigma^2\right) \ , \tag{4.6}$$

where $\boldsymbol{N}^x = \rho^x \boldsymbol{n}^x$. Note that without a prior on the surface orientation, pixels that are infrequently illuminated will be difficult to analyze. Additionally, a prior on the visibility function is necessary to avoid the trivial solution of $\boldsymbol{V} = \boldsymbol{0}$.

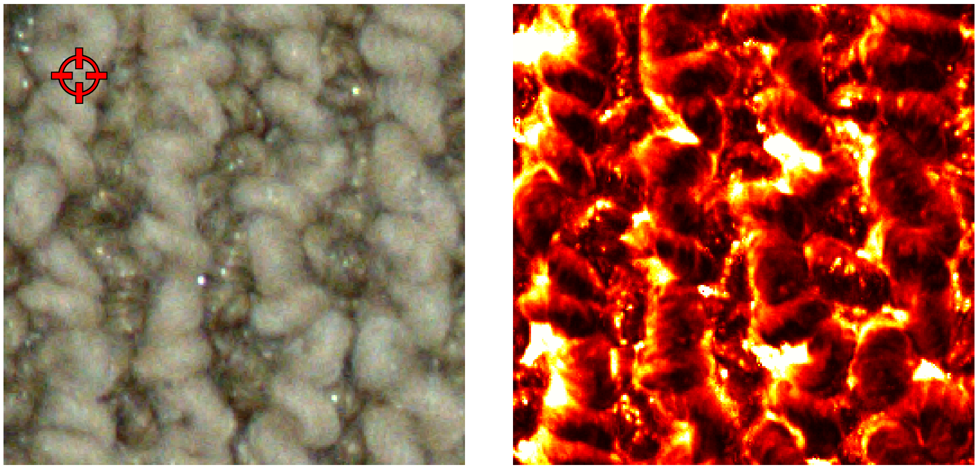
Surface normal prior

We exploit the repetitious nature of texture with a novel nonparametric prior across geometrically similar, but spatially disjoint locations. As noted above, if two pixels share similar intensity profiles, they must share similar surface orientations [68]. We translate this observation into a prior on the surface normals that encourages similar orientation for pixels with similar intensity profiles. We let Ω^x be the set of pixels in the spatially disjoint *intensity profile cluster* of x, and formulate the prior using a kernel density as

$$p(\boldsymbol{N}) = \prod_x \frac{1}{|\Omega^x|} \sum_{y \in \Omega^x} \mathcal{N}\left(\angle(\boldsymbol{N}^x, \boldsymbol{N}^y), h\right) \ , \tag{4.7}$$

where $|\Omega^x|$ is the set's cardinality, $\mathcal{N}$ is the normal density function, with variance h, and $\angle(\cdot,\cdot)$ is the angle between the two normals.

We let Ω^x be the set of locations $\Omega^x = \{y\}$ for which $\mathrm{dist}(x, y)$ is less than some threshold. We determine the intensity profile distance between two locations using only



(a) Texture **(b)** Distance map

Figure 4.3: Example distance map for one pixel. In (b) each pixel is given a brightness proportional to the distance of its intensity profile from that of the targeted pixel of (a) according to Equation 4.8. Note how other downward-facing regions on the knots of carpet have lower (darker) distance values.

the portions of their intensity profiles for which both locations are directly illuminated, i.e. $\{t \mid V_t^x = V_t^y = 1\}$. More precisely, we compute the distance using the cosine similarity of the visibility-clamped intensity profiles

$$\operatorname{dist}(x, y) = 1 - \frac{(\boldsymbol{V}^x \circ \boldsymbol{V}^y \circ \widetilde{\boldsymbol{I}}^x) \cdot (\boldsymbol{V}^x \circ \boldsymbol{V}^y \circ \widetilde{\boldsymbol{I}}^y)}{\|\boldsymbol{V}^x \circ \boldsymbol{V}^y \circ \widetilde{\boldsymbol{I}}^x\| \, \|\boldsymbol{V}^x \circ \boldsymbol{V}^y \circ \widetilde{\boldsymbol{I}}^y\|} \, , \tag{4.8}$$

where $\circ$ denotes element-wise multiplication, i.e. visibility-clamping, $\widetilde{\boldsymbol{I}}_t^x = \boldsymbol{I}_t^x / \rho_x$, and $\| \cdot \|$ indicates the $L2$ norm. By dividing by the estimated albedo, we allow pixels with different albedos, but similar orientations to be part of the same cluster.

Figure 4.3 shows the distance map for one pixel that is located on the downward-facing slope of a carpet knot. As can be seen by the dark regions, this pixel has a low distance value to similarly oriented surface patches. The bright regions, on the other hand, correspond to areas that are not frequently illuminated at the same time, or seldom have similar intensity values.

Visibility function prior

As the light source is waved across the texture, each pixel will remain illuminated for consecutive observations, and in some consistency with the pixels surrounding it. In other words, the visibility function $\boldsymbol{V}$ should be piecewise constant both spatially, as well as temporally. Further, we impose a unary prior on the entire visibility function to avoid the trivial solution of $\boldsymbol{V} = \boldsymbol{0}$. The combined prior is

$$\begin{aligned} p(\boldsymbol{V}) = {} & \exp\big(- \beta_u^V \, \|\boldsymbol{1} - \boldsymbol{V}\|_1 \big) \\ & \exp\big(- \beta_s^V \sum_t \sum_x \sum_{y \in \mathsf{N}_s^x} |V_t^x - V_t^y| \big) \\ & \exp\big(- \beta_t^V \sum_t \sum_x \sum_{y \in \mathsf{N}_t^x} |V_t^x - V_t^y| \big) \ , \end{aligned} \tag{4.9}$$

where N_s^x indicates the four spatial neighbors of pixel x, N_t^x indicates the two temporal neighbors (i.e. the previous and subsequent observations) and $\beta_{\{u,s,t\}}^V$ are the relative weights of the unary, spatial, and temporal components.

Note that setting the spatial prior weight β_s^V quite high, and the temporal prior β_t^V to zero, would result in a similar formulation to that of Chandraker et al. [64] who assume self-occlusions cast large shadows. In our case, however, we use a weak spatial prior to allow for small-scale detail.

Expectation maximization

We estimate the visibility function and surface normals using expectation-maximization. In the expectation step we iterate back and forth, updating each layer while considering the other as a *pseudo-observable* [69]. In both cases, we can find the maximum a posterior by

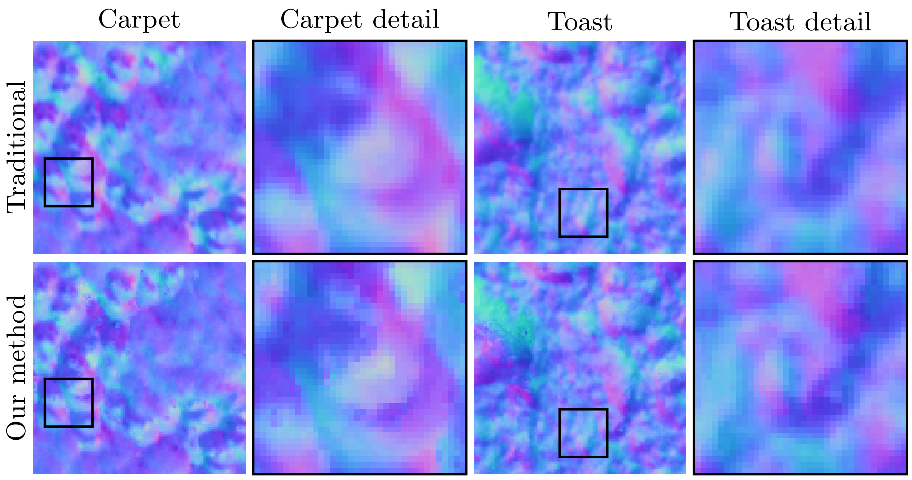


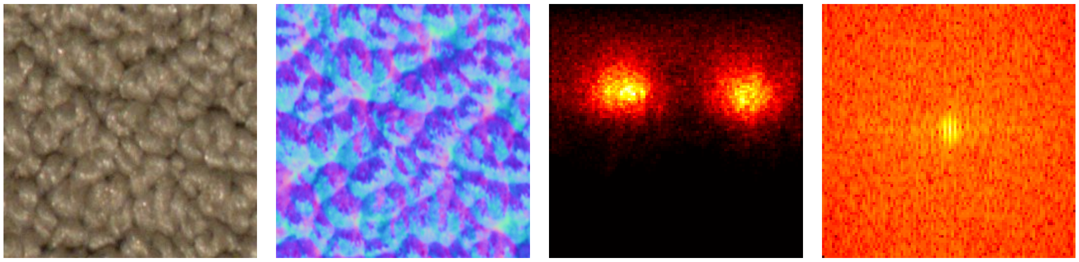
Figure 4.4: Resulting normal field comparison. Note the increased detail of our method.

minimizing its negative log likelihood. In the case of the visibility function, since each node can take on only two possible states (0 or 1), we can approximate the minimum quickly using graph cuts [70].

Due to the nonparametric nature of our surface normal prior, we must make an approximation to avoid intractability. Specifically, we use the previously estimated values for the surface normals in the cluster $\boldsymbol{N}^{y \in \Omega^x}$ in Equation 4.7. This allows each surface normal to be updated independently of the others. To minimize the negative log posterior for each pixel we use a gradient descent method.

In the maximization step, we update the noise variance σ^2 using the maximum likelihood estimator. These two steps, updating the latent values, and maximizing the likelihood, are iterated until convergence which we define as an average change in surface orientation of less than 1°.

In Figure 4.2 we show the result of our method on a synthetic scene of several spheres embedded in a plane. In addition to self-occlusion, as the light passes over the scene, each



(a) Carpet texture **(b)** Normal field **(c)** Histogram **(d)** FFT amp.

Figure 4.5: Steps to form base representation. After computing the surface normals (b), we build a 2D histogram (c) from the polar and azimuth angles of each normal vector. We use the rotation-invariant amplitude of the frequency domain (d) as our base representation.

sphere casts large shadows on the background plane. These shadows result in erroneous normal estimations around each sphere when the traditional approach is used. On the right hand side of the figure we show a series of normal vectors from one slice of the resulting normal fields. Note how our method is able to accurately recover the sharp change from the background plane to the sphere. In Figure 4.4 we compare our result for two real-world textures to the result using the traditional method. Qualitatively, one can see the increased clarity that results from our method over the traditional approach.

4.4 Encoding geometric texture

With the exception of sharp depth discontinuities, the geometry of texture is well described by the normal field. This normal field, however, will change dramatically as the texture is rotated, or moved towards or away from the viewer. An ideal representation of texture geometry will be both rotation-invariant and will encode its scale variability.

4.4.1 Rotation-invariant base representation

In order to obtain a rotation-invariant representation of texture geometry, we first adjust the normal field so that the global surface normal (i.e., the mean normal) is aligned with the positive z-axis. This accounts for small out-of-plane rotations of the texture. As shown in Figure 4.5c, we then form a normalized two-dimensional histogram $h(\phi, \theta)$ from the azimuth angle ϕ and polar angle θ of each surface normal. For our experiments, we quantize the polar and azimuth angles into 100 bins. In Figure 4.5c we show one such histogram in which brighter regions correspond to higher values. We can see that the texture is relatively coarse by observing that it has a highly structured grouping of normals. In this histogram, a rotation about the mean normal corresponds to a shift along the horizontal (azimuth) axis. The final step in our base representation is to apply Fourier transform to the histogram. The amplitude of the transform, shown in log-space in Figure 4.5d, is then a rotation invariant characterization of the normal field.

We define the similarity of two such frequency representations using the Jensen-Shannon divergence [71]. Given two surface textures A and B, we first compute their respective rotation-invariant representations $H_A(u, v)$ and $H_B(u, v)$. We then define their similarity as the Jensen-Shannon divergence of H_A and H_B

$$\mathcal{J}(H_A, H_B) = \frac{1}{2}\left(\mathcal{K}(\widetilde{H}_A, C) + \mathcal{K}(\widetilde{H}_B, C)\right) \ , \tag{4.10}$$

where $\widetilde{H}_A$ and $\widetilde{H}_B$ are the normalized distributions

$$\widetilde{H}_A = \frac{H_A}{\sum\sum|H_A|} \quad \widetilde{H}_B = \frac{H_B}{\sum\sum|H_B|} \ , \tag{4.11}$$

C is their mean, and $\mathcal{K}$ is the Kullback-Leibler divergence.

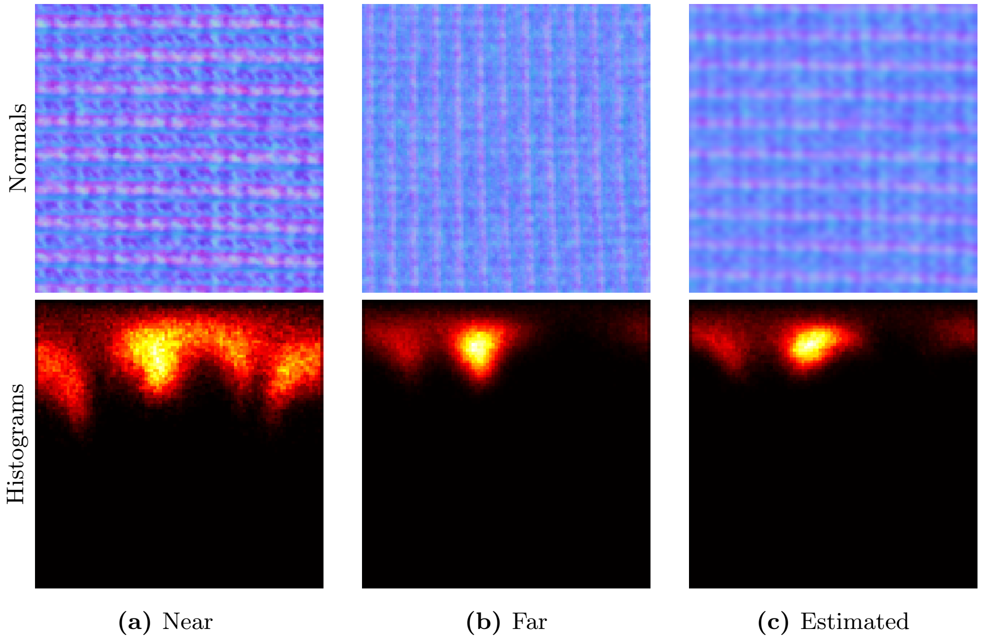
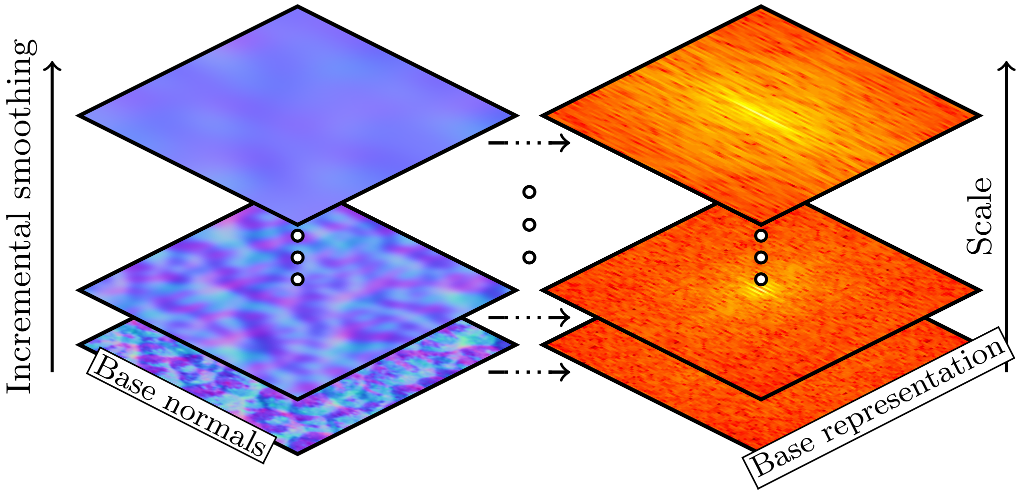


Figure 4.6: Comparison of real and estimated scale. When a texture (in this case, a cloth) is imaged at an increased distance, the apparent normal field becomes more coherent (b). We approximate this by smoothing the original normal field (a). The resulting histogram (c) is quite similar to the histogram of the observed texture.

4.4.2 Scale-space representation

In Figure 4.6 we show the surface normal field for a cloth at two distances. On the far left, the cloth is relatively close to the camera, and in the middle the cloth has been moved twice the distance away. Although the geometry of the cloth has not changed, the number of microfacet surface normals subtended by each pixel has increased. In other words, as scale increases, the appearance of each pixel is determined by an increasing number of surface orientations. In effect, the average surface normal captured by a given pixel is computed across a greater area as scale increases. This observation, also underlying the recent work on geometric scale space of range images [72, 73], motivates our approach to modeling the scale variability of geometric texture.

We describe the scale space of geometric texture by filtering the surface normal field



(a) Normal fields **(b)** Our representation

Figure 4.7: Building of scale-space representation. We represent the scale space of geometric texture with frequency histograms (b) built from the polar histogram (not shown) of the incrementally smoothed normal field (a).

with Gaussian kernels of increasing standard deviation. In Figure 4.6 we illustrate the result of this approach by comparing the resulting normal histogram (c) with that of the texture at a greater distance (b). The result is quite similar to the observed histogram.

Figure 4.7 summarizes the creation process of our scale-space representation. Starting in the lower left, we begin with the observed normal field of texture. As shown in (a), we then smooth this normal field at small intervals. At each interval, the polar histogram is formed and its Fourier transform is computed (b). This process is repeated until the normal field is sufficiently coherent (has an angular standard deviation of 2°). Note that as this happens, the polar histogram becomes nondescript, and corresponding frequency image flattens out, as can be seen at the top of the stack. Smooth textures, whose normal fields are already quite coherent, require few scales to describe their scale variability, while coarse textures will have larger stacks.

4.5 Classification

We evaluate the effectiveness of our rotation-invariant scale-space representation of texture geometry by performing classification on a new database.

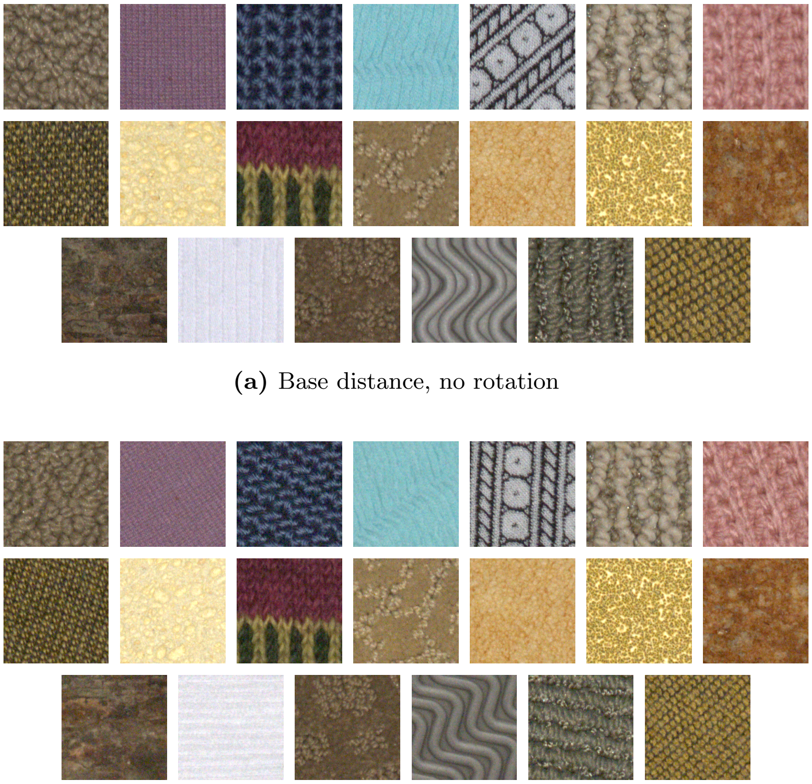
4.5.1 Classifying a query texture

When a query texture is imaged, and its normal field is estimated, classification is performed on its frequency amplitude histogram. In order to find the correct texture, we must also find the correct scale. One approach would be to compute the similarity (Equation 4.10) between the query texture and every scale of every database texture. Needless to say, this process quickly becomes too computationally expensive as the number of textures and scales increases. To more quickly estimate the likely textures we first compare the total energy of the query texture to that of each database texture at each scale. The total energy is computed as the sum of the unnormalized frequency amplitudes.

4.5.2 Geometric texture database

As depicted in Figure 4.8, we introduce a new database that covers 20 textures at different distances, with different in-plane and out-of-plane rotations[1]. To our knowledge, this is the only public database that offers multiple distances for each texture in addition to multiple in-plane and out-of-plane rotations. Our database, which contains stochastic as well as regular textures, includes organic materials such as bark and toast, woven materials such as cloth crocheted fabric, and synthetic materials such as a carpet and sandpaper. For practical reasons, we did not construct a lighting apparatus for each distance. Instead, each dataset was lit by manually waving a light source around in a spiral pattern like the one shown in Figure 4.1b. The lighting direction was estimated using a reflective sphere.

[1] This database is available on-line at `http://cs.drexel.edu/~kon/texture`.



(a) Base distance, no rotation

(b) Assorted distances and rotations

Figure 4.8: Select database samples. Each of our 20 textures is imaged with 30 combinations of orientation and distance.

Each dataset consists of approximately 65 images. Every effort was made to keep the distance from the subject constant, but some amount of light attenuation will necessarily be present. To address this, each image is white-balanced using the light source direction and a white paper in the scene. Since each texture is imaged with more than 2,000 different combinations of orientation, distance, and lighting direction, it is also useful for appearance modeling.

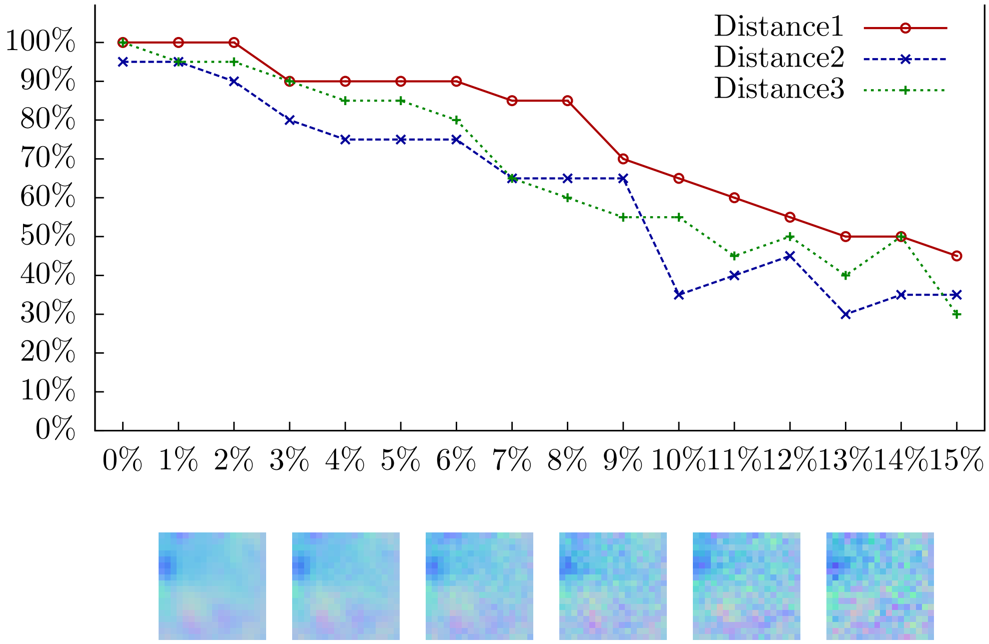


Figure 4.9: Recognition rates under increasing Gaussian noise. The horizontal axis is the standard deviation of Gaussian noise added to the normal field. A sample region is shown for every third level of noise to illustrate the degree of corruption. Note that high accuracy is achieved even with significant corruption.

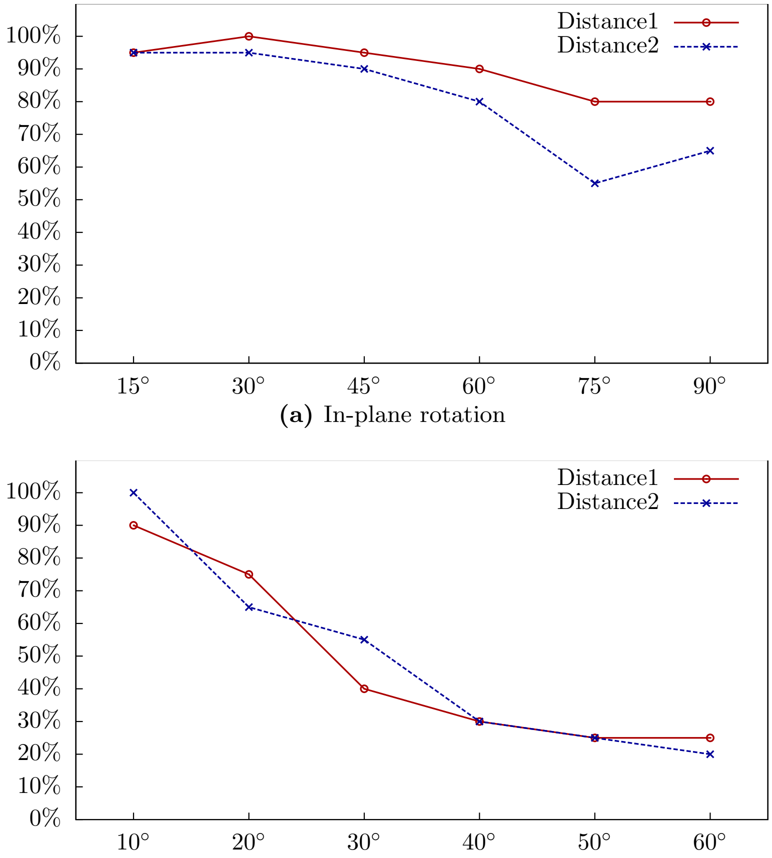
4.6 Experimental validation

Distance

For each texture in the model database, acquired at a distance D1, we acquired additional real data to be used as query instances for classification at two increasing distances D2 and D3. All 20 query textures at D2 were classified correctly, while 19 were correctly classified at D3. These are the the starting points in the top left of Figure 4.9.

Noise

Figure 4.9 shows the robustness of our method to increasing levels of Gaussian noise in the normal field. The horizontal axis corresponds to the standard deviation of the Gaussian noise applied to each dimension of normal field. Below the graph are sample patches of a normal field at every third level of noise corruption. The fourth sample patch shows the



(a) In-plane rotation

(b) Out-of-plane rotation.

Figure 4.10: Recognition rates for rotations Out-of-plane rotations expose previously unseen structures, causing novel normal fields which are more challenging to classify. Although recognition rates decline slightly at increased angles, high accuracy is persistent.

high degree of noise at $\sigma = 9\%$. As shown in red, at this level 70% of the original textures are classified correctly. The blue and green lines correspond to the increased distances of D2 and D3, respectively. Since each surface normal captures an increased amount of surface area at these distances, noise has a greater effect.

In-Plane Rotation

At distances D1 and D2 we rotated each texture 6 times in increments of 15°, stopping at a rotation of 90°. Moving beyond 90° would be redundant, since the same result can be achieved by rotating the images 90°. Overall, at distance D1 we correctly classify 88% of

Out-of-plane rotation angle	In-plane rotation angle						
	15°	30°	45°	60°	75°	90°	∑
10°	**3/3**	**5/5**	**3/5**	**3/4**	**1/1**	**1/2**	**16/20**
20°	**4/5**	**2/2**	**1/2**	**1/2**	**2/6**	**1/3**	**11/20**
30°	**0/4**	**3/4**	**1/2**	**1/2**	**2/4**	**4/4**	**11/20**
∑	**7/12**	**10/11**	**5/9**	**5/8**	**5/11**	**6/9**	**38/60**

Table 4.1: Recognition rates for combinations of in-plane and out-of-plane rotations. For quick inspection, higher recognition rates are given a brighter background coloring. The last row and column are sums. Note the high success rates for the top row and upper left region.

the 120 query textures, and at distance D2 we correctly classify 81% of the query textures. In Figure 4.10a we show the results for each rotation increment.

Out-of-Plane Rotation

At distances D1 and D2 we performed out-of-plane rotations as well. These were done in increments of 10°, stopping at 60°. The results are summarized in Figure 4.10b. Out-of-plane rotations present a unique challenge in that the rotations expose geometry that was previously hidden. In essence, out-of-plane rotations present novel textures as the side of the original texture becomes visible.

Arbitrary Rotations

At distance D3 we tested 3 arbitrary combinations of in-plane and out-of-plane rotations for each texture. Table 4.1 shows the results of all 60 tests. Each row corresponds to an increased out-of-plane rotation angle, while each column corresponds to an increased in-plane rotation angle. Each entry in the table shows the total correct classifications and the number of experiments run with that combination of in-plane and out-of-plane rotation angles as a fraction. The shading of each cell indicates the overall success rate. Note that successful cases tend to have lower out-of-plane rotations, as expected.

4.7 Summary

In this work we have directly studied the geometry of texture. Through careful analysis of its two key characteristics—self similarity and scale variability—we have derived a photometric stereo method specifically tailored to exploit the repetitive nature of texture geometry, and have introduced a compact representation for the scale space of geometric texture. To evaluate our methods we have also introduced an extensive new texture database. Experimentally, we have shown that our compact scale-space representation is highly discriminative. In doing so, we have provided a foundation for the application texture geometry to longstanding computer vision problems, such as scene and material understanding. With affordable geometric sensors on the rise, geometric approaches will become increasingly prolific. Our classification method may be applied to normal fields produced by any method or sensor, opening up the range of applications.

Chapter 5: Geometry recovery in the wild

5.1 Introduction

When we look at the world around us, we do not see a series of 2D images. We see geometry, and photometry. By perceiving geometry, we enable ourselves to interact with the world, and by perceiving photometry, we can infer physical properties.

Our world is full of objects made of different materials situated in complex environments. The combined effect of this interaction, though beautiful, makes decoupling appearance into geometry and photometry extremely challenging. Past work on geometry estimation has therefore focused on compressing this complexity to fit simple models. Illumination is often controlled and assumed to be point light sources, and reflectance is often assumed to be Lambertian, mirrored or dichromatic.

Though such assumptions give rise to beneficial properties that aid geometry estimation, they limit us to a small slice of real-world materials or force us into the darkroom where we can control lighting. In this chapter we relax both material and illumination assumptions to fully bring shape (and reflectance) estimation into “the wild”. Our goal is to estimate what we cannot directly (and passively) acquire—the shape and reflectance of the object—from what we can—one or more images of the object and a panorama of the illumination environment.

Certainly it is true that when an object with a non-trivial reflectance is situated in a non-trivial illumination environment we can expect complex appearance. The appearance of each point on the object, however gives strong clues about its orientation. Unique portions of the illumination environment (such as the sun, or stained glass) act analogously to the point light sources of past work in that they constrain the space of possible orientations to those that reflect them. The strength of this constraint, however, depends on the reflectance, the illumination, and what part of the scene is being reflected. Points reflecting unique regions of the scene are tightly constrained while those reflecting less salient regions maybe quite weekly constrained.

If only a single image is available, we address the ambiguity in weakly constrained regions with careful priors designed to propagate the information from tightly constrained regions. If multiple images are available then such weak constraints become strong when additional viewpoints corroborate a tighter range of possible orientations.

In either case an inherent ambiguity exists tightly coupling shape and reflectance. The shape of the object cannot be deduced without knowledge of the reflectance which modulates the illumination. On the other hand, the reflectance itself cannot be guessed without understanding what scene components are being reflected. Our overall approach is therefore to estimate the two jointly.

We leverage the Directional Statistics BRDF model for reflectance, and utilize a normal field or triangle mesh to represent geometry in the single and multiview cases respectively. By keeping one fixed as the other is estimated, and utilizing novel constraints we may recover both without enabling either to absorb the error of the other.

We evaluate our methods extensively on synthetic and real world data for which we have acquired ground truth. The results show that our approaches enable accurate shape and real-world reflectance estimation under complex natural illumination.

5.2 Related work

Geometry estimation, whether from a single image, or multiple images, is a cornerstone of computer vision research. Instead of giving a full survey of these longstanding problems, we will focus on those methods that have important similarity to our own. Interested readers can refer to surveys by Durou et al. [74], Zhang et al. [75] and Seitz et al. [76] for more context.

5.2.1 Single viewpoint geometry estimation

The problem of single view geometry estimation has traditionally taken place in the darkroom where the illumination conditions can be controlled. Goldman et al. [77], for example, moved beyond the Lambertian reflectance assumption to the more general parametric Ward model. By capturing many images as a light source is moved around, sufficient information can be gathered to estimate the reflectance parameters and surface orientations. For more general, isotropic reflectances, Alldrin et al. [23, 78] showed that by structuring the illumination densely around the viewing direction, isocontours can be extracted that give depth information regardless of the actual (isotropic) BRDFs of the object. Unfortunately, no such structure can be assumed about natural illumination. Indeed appearance is tightly coupled to the combined role of reflectance and shape. We therefore seek to jointly estimate the reflectance along with the object geometry.

Hertzmann and Seitz tackle the problem with a different assumption [79]. By assuming that they additionally have a sphere of the same material as the target object, they can work with much more general reflectance types. The sphere then serves as a measured reflectance map. By comparing the sequence of appearances of a surface patch on the target object with those of all the points on the reference sphere, the orientation for the surface patch can be deduced. The downside, however, is that a sphere of the same material needs to be acquired (or painted). By jointly estimating the reflectance, we move away from requiring a known reflectance.

There has been some work on shape estimation in natural illumination though the focus has been on Lambertian reflectance. Huang and Smith [80], and Johnson and Adelson [24] observe that under natural illumination, the appearance of a Lambertian object takes a parametric form despite the inherently non-parametric nature of the illumination environment. By representing the illumination using the spherical harmonics model [81], a parametric reflectance map can be computed that directly links appearance to orientation. Huang and Smith [80] also showed that the illumination parameters themselves can be estimated. Barron and Malik [25, 26] went further, enabling estimation of outlier non-Lambertian effects (highlights). When the reflectance is not assumed to be Lambertian, however, the reflectance map stays non-parametric, making such approaches inapplicable.

Attention has also been paid to the other extreme form of reflectance–the purely specular. Adato et al. [82] observe the flow of the reflected, yet unknown, illumination environment for a known relative movement of the environment. In order to effectively rotate the environment, however, this method requires that the relationship between the camera and the object be fixed, and be able to move together. Tappen [83] observes that the local curvature of mirrored surfaces result in characteristic patterns (such as the curved lines of distorted buildings and trees). Without knowing the illumination, such patterns give clues about the curvature that can then be used to recover the shape. For general BRDFs, however, such detailed analysis of the appearance is not feasible as the reflectance will smooth away the edges of the scene which help encode curvature.

Between the extremes of Lambertian and purely specular, we have the wide range of real world reflectances that surround us. Here, neither a direct nor parametric link between illumination and orientation exists. In this chapter we show that by jointly estimating the reflectance, nonparametric distributions of likely orientations can be extracted for each pixel. In the single image case, we show how this complex problem becomes tractable through priors that extend the influence of the sparse but salient orientation clues that are reflected from the lighting environment.

5.2.2 Multiple viewpoint geometry estimation

A single image is inherently limited in its ability to capture the geometry of an object. As such, the problem of multiple viewpoint geometry estimation has also received considerable attention. Although the connection to the single viewpoint case is intuitive, the problem itself is not, and there are many ways to formalize the relationship.

Lambertian objects exhibit another helpful property, that of viewpoint independent appearance. A surface patch that has a certain appearance when viewed from one direction will appear the same when viewed from another direction. This simple observation gives way to the notion of *photometric consistency*. By dividing up the 3D space that the object sits in into voxels, we can then test if a voxel contains the actual object surface by checking if its appearance is consistent throughout the various observation images [84–86]. In order to build robustness to changes in radiance this notion was later extended to patch-based comparisons by Pons et al. [87]. Jin et al. [88, 89] moved the concept beyond Lambertian reflectance to the Ward model. They note that this more general reflectance still exhibits strongly constrained appearance variation. They then exploit this by measuring (and constraining) the radiance tensor field across many (~ 40) images. In this chapter we go further by working with arbitrary isotropic BRDFs, and by working under natural illumination.

Past work has approached the problem of arbitrary reflectance by extending the notion of consistency to *orientation consistency*. By leveraging past work on single-viewpoint geometry estimation, each viewing direction can be converted into an analogous geometry observation. Hernàndez et al. [90], for example, use controlled lighting to convert each observation location into a reliable geometry observation in the form of a surface normal field. Similarly, Treuille et al. [91] use a reference sphere of the same material to extract normal fields for each observation location. The goal for these methods is then to estimate a full 3D model that is consistent with the newly created geometry observations. Our multiview framework has a similar motivation, but operates with the highly ambiguous, non-parametric orientation distributions that result in natural illumination for non-trivial reflectances. In other words, instead of having reliable geometry observations in the form of normal fields we have non-parametric multi-modal distributions of possible orientations. We introduce a framework for compactly representing this complexity to perform full shape (and reflectance) estimation.

5.3 Bayesian shape and reflectance estimation

The appearance of an object is due to the illumination, viewing conditions, shape of the object, and its reflectance. We assume the illumination is known but uncontrolled natural illumination $\boldsymbol{L}$, the object material has an isotropic reflectance function, and that it has been segmented from the background. We also assume that we have one or more images $\boldsymbol{\mathcal{I}} = \{\boldsymbol{I}^1, \ldots, \boldsymbol{I}^M\}$, $M \geq 1$, from a calibrated camera. These assumptions can be met using existing work (SfM may be sufficient if the surrounding environment is feature-rich).

Our primary contribution is a probabilistic framework for estimating the remaining components—the geometry $\boldsymbol{G}$, and reflectance $\boldsymbol{\Psi}$. We formulate this as the maximum a posteriori (MAP) estimate of the posterior distribution

$$p(\boldsymbol{G}, \boldsymbol{\Psi} | \boldsymbol{\mathcal{I}}) \propto p(\boldsymbol{\mathcal{I}} | \boldsymbol{G}, \boldsymbol{\Psi}) p(\boldsymbol{G}) p(\boldsymbol{\Psi}) \ , \tag{5.1}$$

where the likelihood $p(\boldsymbol{\mathcal{I}} | \boldsymbol{G}, \boldsymbol{\Psi})$ quantifies how consistent the geometry and reflectance are with the observation(s), and the priors $p(\boldsymbol{G})$ and $p(\boldsymbol{\Psi})$ encode practical constraints.

```
Data: Illumination $\boldsymbol{L}$ and images $\mathcal{I}$
Result: Geometry $\boldsymbol{G}$ and reflectance $\boldsymbol{\Psi}$
$\boldsymbol{G} \leftarrow$ initial geometry estimate;
while not converged do
    update $\sigma$ of Eq. 5.3;
    while not converged do
        update $\boldsymbol{\Psi}$ using $\boldsymbol{L}$, $\mathcal{I}$, and the current $\boldsymbol{G}$;
        update $\boldsymbol{G}$ using $\boldsymbol{L}$, $\mathcal{I}$, and the current $\boldsymbol{\Psi}$;
    end
end
```

Algorithm 1: **Joint optimization of reflectance and shape** After initialization, the algorithm alternates between estimating the image formation variance and estimating the shape and reflectance, which themselves are estimated in an alternating fashion.

Algorithm 1 gives an overview our iterative optimization framework, which is modeled after expectation maximization. After initialization, an outer loop alternates between updating the likelihood variance (discussed below) and updating the geometry and reflectance. When updating these two components one is kept fixed while the other is updated, alternating back and forth until convergence.

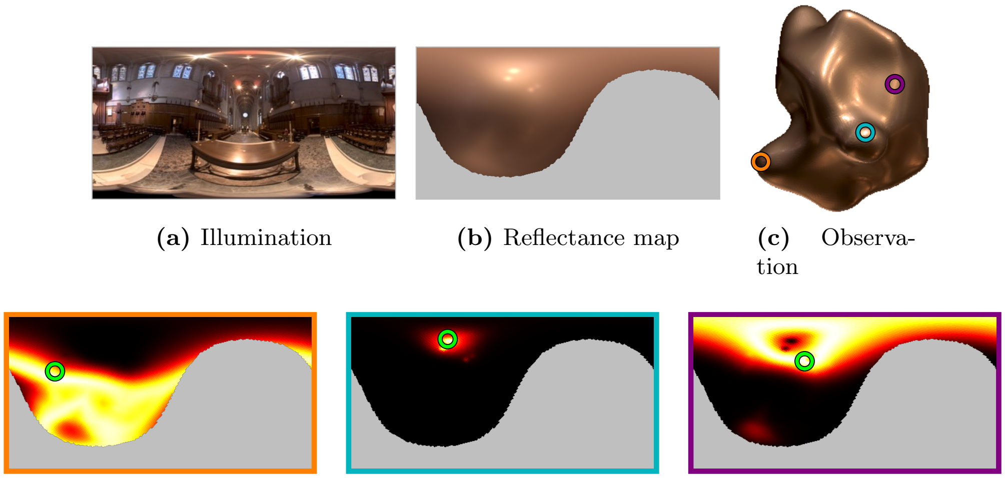
5.4 A single image

Let us first fully investigate what can be learned from a single image. First we will investigate how shape can be deduced from reflectance, then we will discuss how reflectance can be deduced from shape. We will then show how to soundly initialize this iterative process.

5.4.1 Shape from reflectance

Image formation likelihood

Let us begin with a single pixel $\boldsymbol{I}_x$ in the image $\boldsymbol{I}$. For example, consider the pixel of Figure 5.1c circled in orange. The appearance of this pixel is due to the reflectance $\boldsymbol{\Psi}$, the illumination environment $\boldsymbol{L}$, and the underlying orientation $\boldsymbol{N}_x$ of the corresponding



(a) Illumination

(b) Reflectance map

(c) Observation

(d) Likelihoods for color-coded example pixels (brighter is more likely)

Figure 5.1: Orientation likelihood spherical panoramas The illumination (a) and reflectance combine to form a complex reflectance map (b). When this is compared with the observed appearance of a pixel in an observation (c), we arrive at a non-parametric distribution of orientations (d). The green circles denote the true orientation of the corresponding pixels (sorted left to right, and color coded).

surface point, with some added Gaussian noise of uniform variance,

$$\boldsymbol{I}_x = \boldsymbol{E}(\boldsymbol{\Psi}, \boldsymbol{L}, \boldsymbol{N}_x) + \mathcal{N}(\boldsymbol{0}, \sigma^2) \ . \tag{5.2}$$

The likelihood thus takes the form of a Gaussian centered on the predicted irradiance $\boldsymbol{E}_x$. Here we use the log-intensities to remain sensitive to subtle detail as well as highlights,

$$p(\boldsymbol{I}_x|\boldsymbol{N}_x) = \mathcal{N}\big(\ln(\boldsymbol{I}_x)|\ln(\boldsymbol{E}_x), \sigma^2\big) \ . \tag{5.3}$$

The orientation $\boldsymbol{N}_x$ determines what hemisphere of light will be modulated by the reflectance and integrated to form the appearance. As shown in Figure 5.1b, since the predicted irradiance is a function of $\boldsymbol{N}_x$ (which itself can be expressed in 2D spherical coordinates $\boldsymbol{N}_x = (\theta_x, \phi_x)$), we can visualize it as a 2D spherical panorama. Note that only the half of the image corresponding to the camera-facing hemisphere is filled in, while the self-occluded half is shown in light gray.

The likelihood may be visualized similarly, by computing Equation 5.3 for each orientation of a spherical panorama. Three examples are shown in Figure 5.1d where brighter values correspond to higher-probability orientations. Note how examples 1 and 3 have no clear minimum; the true orientation of the underlying surface point (which is indicated with a green circle) can seldom be directly inferred by appearance.

The overall likelihood can then be expressed over all pixels of the image

$$p(\boldsymbol{\mathcal{I}}|\boldsymbol{G}) = p(\boldsymbol{I}|\boldsymbol{G}) = \prod_{x \in \Omega} p(\boldsymbol{I}_x|\boldsymbol{N}_x) \ , \tag{5.4}$$

where, in the single image case, the overall shape of the object $\boldsymbol{G}$ is expressed as a set of surface normals $\boldsymbol{G} = \{\boldsymbol{N}_x\}$ for each pixel x of the object Ω.

Orientation constraints

In the center of figure 5.1d we see a single bright spot in the distribution indicating that the appearance of this pixel, which lies in a highlight, can only be explained by a small range of orientations. This occurrence, though rare, gives valuable context to the area surrounding the well-determined pixel. We introduce two spatial priors to help propagate this information. We also utilize the occluding boundary as a strong unary prior p_b on pixels at the edge of the object. The spatial priors p_g and p_s are formulated pairwise,

$$p(\boldsymbol{G}) = \prod_{x \in \Omega} \left(p_b(\boldsymbol{N}_x) \prod_{y \in \mathrm{ne}(x)} p_g(\boldsymbol{N}_x, \boldsymbol{N}_y) p_s(\boldsymbol{N}_x, \boldsymbol{N}_y) \right) \ . \tag{5.5}$$

Occluding boundary prior As first explored by Ikeuchi and Horn [92], surface patches on the occluding boundary of smooth objects must be oriented orthogonally to the viewing direction. Since the area subtended by each boundary pixel may include a range of orientations, not all of which will be orthogonal to the viewing direction, we formulate this prior as a tight, but non-singular, distribution,

$$p_b(\boldsymbol{N}_x) = \begin{cases} \mathcal{N}(\arccos^2(\boldsymbol{N}_x \cdot \boldsymbol{B}_x), 0.1) & \text{if } x \in \boldsymbol{B} \\ 1 & \text{otherwise} \ , \end{cases} \tag{5.6}$$

where $\boldsymbol{B}$ is a list of the orthogonal vectors for pixels on the occluding boundary of the object.

Reflected gradient prior Our first spatial prior ensures that the resulting gradient is the same as the observed image gradient

$$p_g(\boldsymbol{N}_x, \boldsymbol{N}_y) \propto \exp - \Big\{ \|\boldsymbol{E}(\boldsymbol{\Psi}, \boldsymbol{N}_x) - \boldsymbol{E}(\boldsymbol{\Psi}, \boldsymbol{N}_y)\| - \|\boldsymbol{I}_x - \boldsymbol{I}_y\| \Big\}^2 \ , \tag{5.7}$$

where $\|\cdot\|$ is the Euclidean norm.

Smoothness prior Although the gradient prior gives sufficient context to pixels in textured areas, in many cases, the reflectance map contains many areas with a similar appearance. Our smoothness prior is designed to allow small changes in orientation (angles less than $\pi/3$) while strongly penalizing sharp changes in orientation. To do so, we formulate this as a logistic function,

$$p_s(\boldsymbol{N}_x, \boldsymbol{N}_y) \propto \Big(1 + \exp\big\{ - s(\cos^{-1}(\boldsymbol{N}_x \cdot \boldsymbol{N}_y) + t\big\}\Big)^{-1} \ , \tag{5.8}$$

where $t = \pi/3$ is the threshold between weak and strong penalization and $s = 10$ is the speed of the transition. When well-determined points are present in the scene, this prior helps ensure that the geometry in the intermediate area varies smoothly.

5.4.2 Reflectance from shape

Now we will describe our method for estimating the reflectance using the image of the object, the illumination environment, and the current geometry estimate $\boldsymbol{G}$ as input.

The directional statistics BRDF model

To model the reflectance function, we adopt the Directional Statistics Bidirectional Reflectance Distribution Function (DSBRF) model, introduced by Nishino [93, 94] and later extended by Lombardi and Nishino [95] to estimate reflectance in natural illumination. The model offers a compact representation of isotropic BRDFs and is naturally paired with a simple, data-driven prior.

Using a linear camera, the irradiance $\boldsymbol{E}(\boldsymbol{\Psi}, \boldsymbol{L}, \boldsymbol{N}_x)$ (of Equation 5.2) is

$$\boldsymbol{E}_x = \int \varrho(t(\omega_i, \omega_o); \boldsymbol{\Psi})\mathbf{L}(\omega_i) \max(0, \boldsymbol{N}_{\mathbf{x}} \cdot \omega_i) d\omega_i, \tag{5.9}$$

where t is a function that transforms the incoming ω_i and outgoing ω_o angles into the alternate BRDF parameterization variables θ_d and θ_h. The reflectance function is expressed as a sum of lobes

$$\varrho^{(\lambda)}(\theta_d, \theta_h; \kappa^{(\lambda)}, \gamma^{(\lambda)}) = \sum_r \exp\left\{\kappa^{(r,\lambda)}(\theta_d) \cos^{\gamma^{(r,\lambda)}(\theta_d)}(\theta_h)\right\} - 1\,, \tag{5.10}$$

where the halfway vector parameterization (i.e., (θ_h, ϕ_h) for the halfway vector and (θ_d, ϕ_d) for the difference vector) [96] is used. $\kappa^{(\lambda)}$ and $\gamma^{(\lambda)}$ are functions that encode the magnitude and acuteness of the reflectance, respectively, of lobe r along the span of θ_d for a particular color channel λ. These curves are modeled as a log-linear combination of data-driven basis functions,

$$\kappa^{(r,\lambda)}(\theta_d) = \exp\left\{ b_\mu(\theta_d; \kappa, r, \lambda) + \sum_i \psi_i b_i(\theta_d; \kappa, r, \lambda) \right\},$$
$$\gamma^{(r,\lambda)}(\theta_d) = \exp\left\{ b_\mu(\theta_d; \gamma, r, \lambda) + \sum_i \psi_i b_i(\theta_d; \gamma, r, \lambda) \right\},$$

where b_μ is the mean basis function, b_i is the i^{th} basis function, and ψ_i are the DSBRDF coefficients. We may compute these basis functions from a set of measured reflectance functions using functional principal component analysis (FPCA).

Probabilistic reflectace estimation

In order to estimate the parameters $\boldsymbol{\Psi}$ we continue with our probabilistic formulation of Equation 5.1. Here, the likelihood is the same as above, though the geometry, and hence the per pixel normals $\{\boldsymbol{N}\}$, are kep fixed

$$p(\boldsymbol{\mathcal{I}}|\boldsymbol{\Psi}) = p(\boldsymbol{I}|\boldsymbol{\Psi}) = \prod_{x \in \Omega} \mathcal{N}\big(\ln(\boldsymbol{I}_x) | \ln(\boldsymbol{E}_x), \sigma^2\big) , \tag{5.11}$$

where, again, Ω is the set of all pixels x of the object, $\boldsymbol{I}_x$ refers to the appearance of the pixel, and $\boldsymbol{E}_x$ refers to its predicted irradiance.

We utilize the prior by Lombardi and Nishino [95], which encourages the coefficients $\psi_i \in \boldsymbol{\Psi}$ of the eigen-functions to be within the distribution of observed reflectances,

$$p(\boldsymbol{\Psi}) \sim \mathcal{N}(0, \beta_\Psi \boldsymbol{\Sigma}_\Psi), \tag{5.12}$$

where the covariance $\boldsymbol{\Sigma}_\Psi$ is computed from the MERL database [97], and the scalar β_Ψ controls the prior strength.

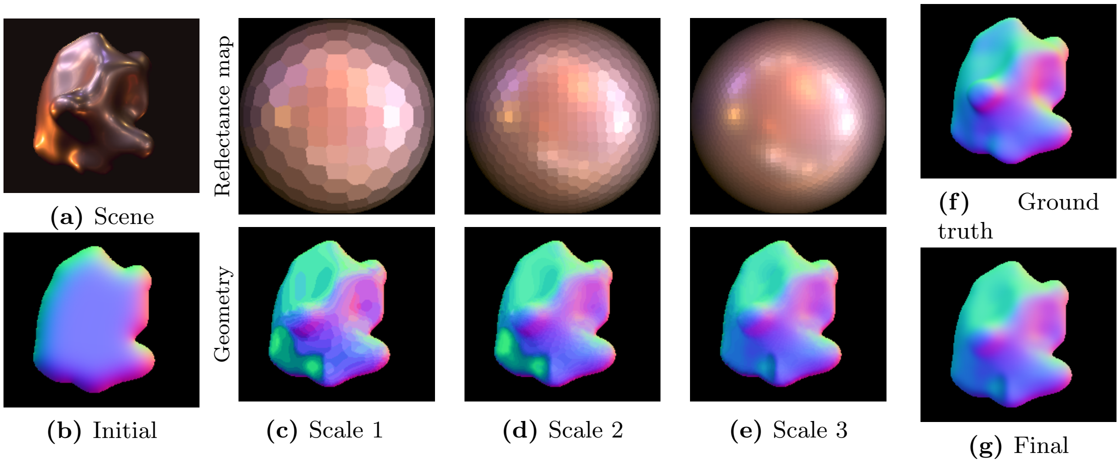


Figure 5.2: Multi-scale geometry estimation. We partition the domain of possible surface orientations into a geodesic hemisphere. By incrementally increasing the resolution of this hemisphere we are able to avoid local minima. Our final step is a gradient-descent based minimization to fine-tune the result.

5.4.3 Implementation and optimization

Since neither the reflectance nor the geometry can be optimized without knowledge of the other, we employ an iterative, expectation maximization (EM) framework in which one is held fixed as a “pseudo-observable” while the other is estimated in the maximization step, while the expectation step estimates the Gaussian likelihood variance.

5.4.4 Initial estimation

We begin our optimization with a naive geometry estimation. Without an estimate for the reflectance, the likelihood of Equation 4.6 and the reflectance gradient prior of Equation 5.7 are meaningless. We may still, however, use the smoothness and occluding boundary priors. As shown in Figure 5.2b, the result is a normal field that resembles a soap-bubble. Although this is a crude estimate of the object geometry, it serves as a reasonable starting point and allows us to bootstrap the iterative EM optimization.

5.4.5 Multi-scale geometry estimation

As we have noted, the primary challenge in this problem stems from the non-parametric nature of the reflectance map. Even if a pixel has a unique appearance, the reflectance map is rarely smooth enough that a simple iterative optimization scheme will find this optimal value. To address this, we introduce a novel global optimization algorithm. As shown in Figure 5.2, we limit the space of possible surface orientations to a set of evenly-distributed orientations on the geodesic hemisphere. In this figure the reflectance map is rendered as a hemisphere instead of a half-filled spherical panorama to conserve space.

There are several benefits to this formulation. First, by limiting the surface orientations to a finite set, we enable the use of rapid global optimization approximation methods. And second, the effect of an inaccurate reflectance estimation is minimized. An inaccurate reflectance estimate may cause the likelihood of Equation 4.6 to have a unique minimum that is incorrect. By limiting the possible orientations, we force each pixel to align with the *region* it is most similar to, while obeying the constraints of our priors. The size of these regions is gradually decreased allowing the priors to play a more subtle role at finer scales.

For a given material estimate, the surface orientations are estimated at each of three scales by incrementally dividing the geodesic hemisphere. To approximate the optimal set of orientations, we use the alpha-beta swap method of Boykov et al. [98]. To help interpolate from one scale to the next, the orientation estimate from the previous scale is first smoothed with a Gaussian kernel before being re-discretized at the next finer resolution. The progression of this optimization scheme is illustrated in Figure 5.2.

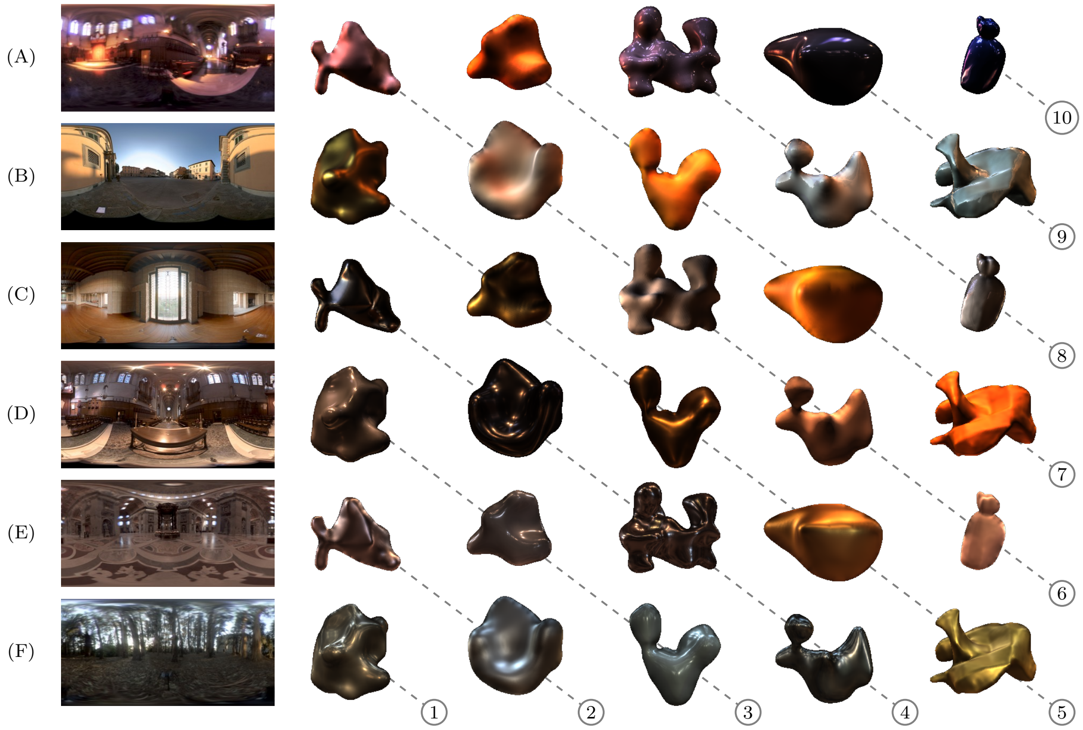


Figure 5.3: Sampling of 600 synthetic images. The bulk of our synthetic data are formed by rendering 10 shapes (5 shown per row) [24] with 10 real-world BRDFs (numbered diagonals) [97] under 6 real-world natural illumination environments (one per row) [99].

5.4.6 Refinement and integrability

At this point we have a good estimate for the object geometry (and reflectance). In order to ensure that the underlying surface in integrable, and to refine the still somewhat coarse orientation estimate, we apply a gradient descent optimization. Similar to Ikeuchi and Horn [92], we perform this optimization in gradient space. The primary difference in our work is due to the DSBRDF. The final shape estimation for our running example is shown in Figure 5.2g. Note that this refinement is only possible because we already have a good estimate through discrete global optimization.

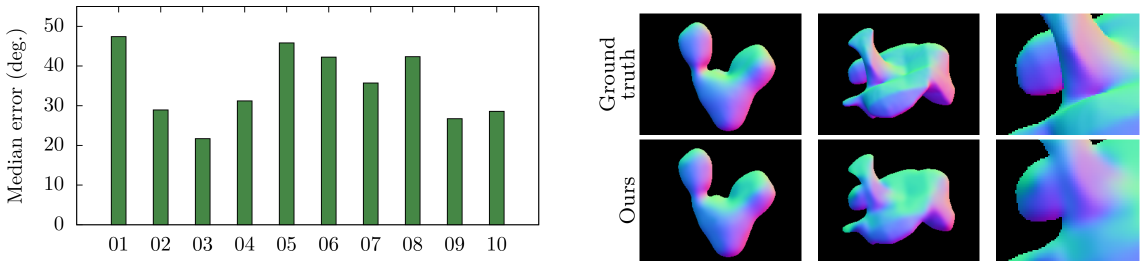
5.4.7 Experimental evaluation

5.4.8 Synthetic images

As illustrated in Figure 5.11 we evaluated our results on 600 synthetic images. Each row of the figure shows 5 of the 10 different shapes from the Blobby Shapes database [24]. Each of these 10 shapes is rendered under 6 different publicly-available real-world illumination environments [99] with each of 10 different real-world measured BRDFs which were chosen from the MERL database [97] to span a wide variety of reflectances. Objects rendered with the same reflectance are shown along each numbered diagonal. The diversity of appearance across each row, and along each diagonal shows how dramatically appearance varies due to changes in material and illumination.

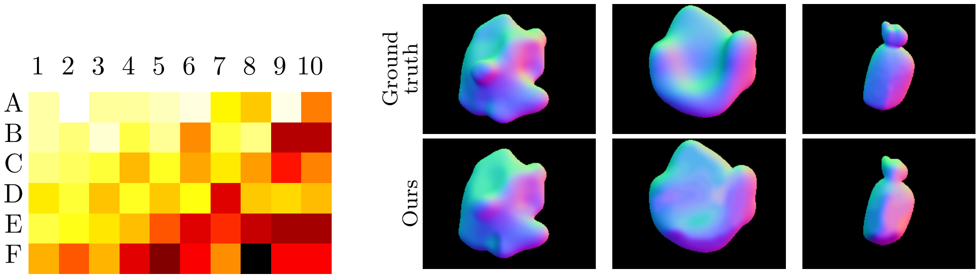
We compute the DSBRDF prior with the ground-truth BRDF omitted. In other words, the ground-truth BRDF is not a part of the training data for the object being analyzed. In total we achieve a median angular error of 24°. The relative root-mean-squared error for the recovered BRDFs ranges from 0.54 to 3.02 though the median value is 1.1. In order to directly compare with the work of Johnson and Adelson [24], we have also briefly tested our method assuming the ground-truth reflectance map is known. In this case we achieve an median angular errors consistently below 15°. This shows the strength of our model to overcome the inherent ambiguities of a non-parametric reflectance map. The increased error when the reflectance is not known is due to the added challenge. We will focus the rest of this discussion on that case.

As shown in Figure 5.4, some shapes are more challenging than others to estimate. In (a) we graph the median angular error, in degrees for each of the 10 shapes averaged over the 60 different lighting and reflectance combinations. In (c) we show the most challenging shape. Note that this shape has several self-occlusions which violate our assumptions of



(a) Geometry error for each blob **(b)** Blob03 **(c)** Blob5 **(d)** Detail

Figure 5.4: Shape accuracy. Shapes with self occlusions, like Blob05 (c) and Blob01 present challenges to our algorithm, while simple shapes like Blob03 (b) are more reliably estimated.

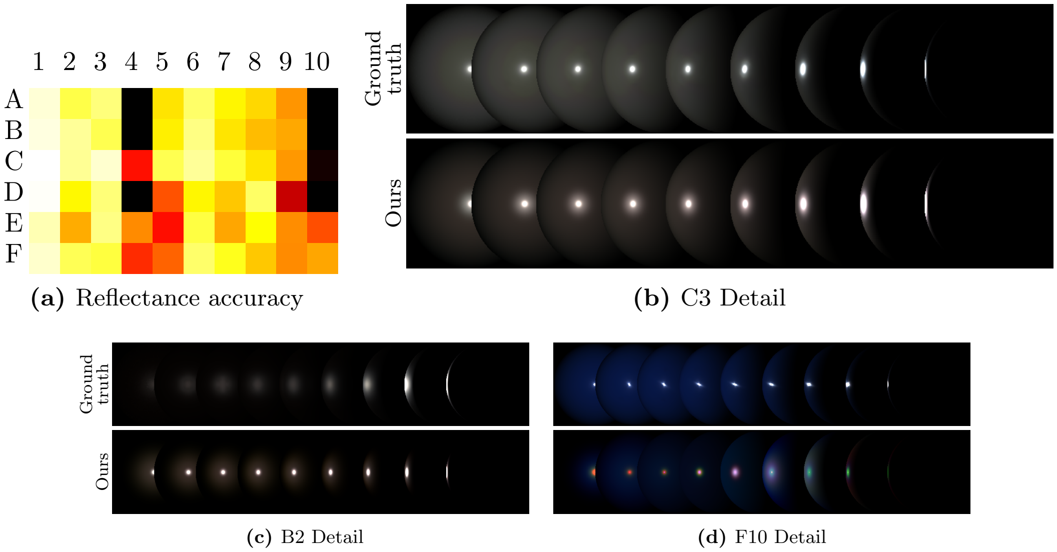


(a) Surface normal accuracy **(b)** B3 Detail **(c)** D6 Detail **(d)** F10 Detail

Figure 5.5: Geometry recovery. The rows and columns of (a) correspond to the labeled illumination maps and reflectances (respectively) in Figure 5.11. The brightest value corresponds to the reflectance and lighting environment combination that yielded the most accurate geometry estimation, with a median error of 14°. The darkest value corresponds to a median error of 49°.

integrability and smoothness. In (b) we show a shape that is easier to estimate. Taken together, these 10 shapes constitute a range of characteristics. For the remainder of our evaluation we will present averages over all 10 shapes.

Figure 5.5 illustrates the accuracy of our geometry recovery for each of the 60 different illumination environment and reflectance combinations averaged over the 10 different shapes. The heat map (a) shows the median angular error with brighter colors used to indicate increased accuracy. In this figure the rows and columns correspond to the labels



(a) Reflectance accuracy

(b) C3 Detail

(c) B2 Detail

(d) F10 Detail

Figure 5.6: Reflectance accuracy. The rows and columns of (a) correspond to the labeled illumination maps and reflectances (respectively) in Figure 5.11. The brightest value corresponds to the most accurate reflectance estimation with a relative RMS error of 0.54. The darkest regions correspond to an error of greater than 3.0.

in Figure 5.11. Several examples are included to illustrate the results.

Some reflectances, such as GRAY PLASTIC (3) and SPECIAL-WALNUT (2) consistently yield more accurate results. These reflectances, which are the most matte, are more successful when large regions of the lighting environment differ strongly from each other. The notable exceptions to this occur in the eucalyptus forest lighting environment (F) and in the St. Peter's environment (E). These two lighting environments have high-frequency texture whose details become nearly uniform under these diffuse reflectances. To test the limits of our method we chose an anisotropic material—WHITE FABRIC (5). As expected, the reflectance estimates and consequently, the geometry estimates for this reflectance are consistently lower than the others.

The outdoor scene from the city of Pisa (B) is shown to facilitate reflectance and geometry estimation. This is likely due to the smooth gradients present in the scene. The result is that, whether specular or diffuse, the reflected appearance is integrating a unique hemisphere of light. On the other hand, environments such as St. Peter's (E) and the Ennis-Brown House (C) have repeated patterns that are less distinctive.

Figure 5.6 illustrates the accuracy of the reflectance estimation. In (b) we show an estimate for GRAY-PLASTIC (3). This reflectance was among the most reliably estimated, due to the smoothly varying appearance resulting from its matte finish. In (d) we show an estimate for BLUE-ACRYLIC (10). This material has subsurface scattering, which our model cannot explain well, causing the global illumination effect to be absorbed in erroneous surface normal and reflectance estimates.

5.4.9 Real-world images

As shown in Figure 5.7, we have also acquired images, and aligned ground-truth geometry for several objects in both out-door and in-door real-world scenes. The ground-truth geometry was acquired using a Canon VIVID 910 light-stripe range-finder. Illumination environments were acquired using a reflective sphere. Although real-world data comes with added sources of noise and inaccuracy, our method is able to recover the shape of the objects quite well in each of the environments.

In each of the four sections of the figure we show the illumination environment along with the image, recovered normal field and ground-truth normals for each of the objects in the scene. The top left section shows a bear figure and a horse figure in a large lobby. The dominant illumination from the overhead skylights is relatively uniform. This leads to some inaccuracy along the top of the objects. The walls of the lobby, however, contain many different points of illumination which contribute to the smoothly varying geometry recovered along the middle of the objects. In this scene the bear has a mean angular error of 24° and the horse has a mean angular error of 17°.

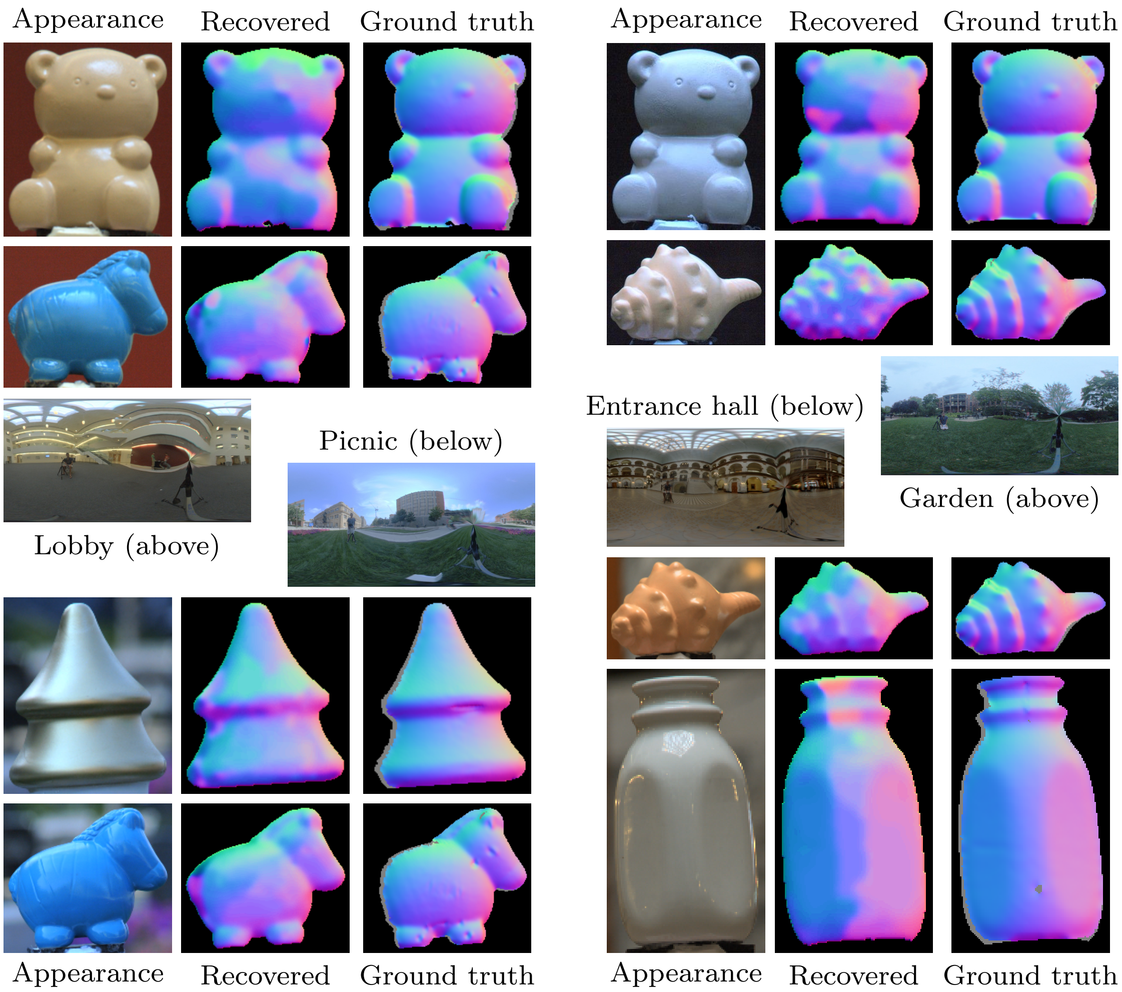


Figure 5.7: Real-world results. We captured several objects in four different natural lighting environments, and aligned ground-truth normal maps. As we discuss in the text, differences in the lighting environments have a clear impact on the accuracy of the recovered geometry.

The garden scene in the top right is illuminated primarily by a cloudy sky. This smoothly varying illumination gives rise to more accurate estimations in the top portions of the objects. In the picnic area scene of the lower left we have a similar environment. This one, however, contains flowers and has more texture in the grass at the bottom of the scene. These two features give extra orientation clues that the garden scene does not have. The effect can be seen most dramatically in the cheek of the bear. Since the lower portion of the garden scene is essentially uniform, points oriented to the left and right have very similar reflected appearances. The objects in the garden scene have mean errors of 19° and 26° whereas the objects in the picnic area scene both have mean angular errors of 17°.

In the entrance hall scene (bottom right) we see the most accuracy along the top of the objects. This is due to variation in the overhead lighting. On the other hand, the pattern in the walls is relatively uniform throughout, giving rise to less accurate detail around the middle of the objects. In this scene the mean error for the objects are 20° for the shell and 16° for the milk bottle.

5.5 Multiple images

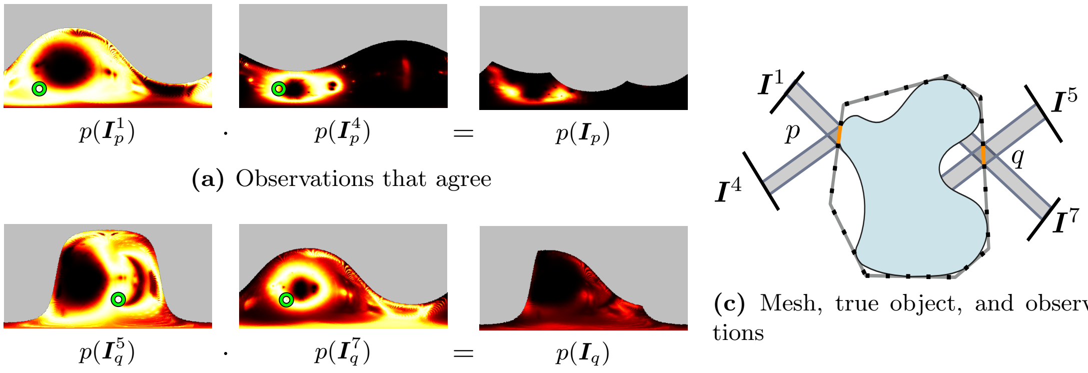
Since a single image is necessarily limited in its expression of geometry, we turn our attention now to the case when multiple images are available.

5.5.1 Shape from reflectance

A unified coordinate frame

When only a single image is available, pixels with ambiguous orientation distributions (like those in Figure 5.1d) must rely on their neighbors to reduce the ambiguity. When multiple images are available, each observation serves as a separate constraint on the distribution of possible orientations. In order to compare observations from different images, however, we must first provide a means to link regions from different images to the same physical location on the object surface.

Figure 5.8 illustrates how a single geometry model can be used to coordinate the observations. On the right of the figure we see the ground-truth object (with a solid boundary) circumscribed by a coarse geometry estimate (dashed). If we take a single point p, and project it into each of the observations images $\boldsymbol{I}^m \in \boldsymbol{\mathcal{I}}$ we may then compute the likelihood



(a) Observations that agree

(b) Observations that disagree

(c) Mesh, true object, and observations

Figure 5.8: Nonparametric orientation consistency When a point on the mesh (dashed) is accurate, the observations will agree (a), resulting in a dense orientation distribution with a clear peak (bright region). When the point is not yet well aligned, the observations will disagree (b), resulting in a flat, near-zero distribution.

density for that point as a product of the separate observations

$$p(\boldsymbol{\mathcal{I}}_p|\boldsymbol{N}_p) = \prod_{m\in\Omega_p} p(\boldsymbol{I}_p^m|\boldsymbol{N}_p) \ , \tag{5.13}$$

where $\boldsymbol{I}_p^m$ is the appearance of the projected point p in image m, and Ω_p is the set of images that can view the point. Note that $p(\boldsymbol{I}_p^m|\boldsymbol{N}_p)$ is identical to the single image Equation 4.6, but for a back-projected surface point.

Two examples are shown in Figure 5.8. In the case on the top (a), the imaged point p is quite close to the true geometry. A direct consequence of that is that the actual imaged appearance is of the same surface point in both $\boldsymbol{I}^1$ and $\boldsymbol{I}^4$. The orientation distributions for these two observations therefore overlap nicely, and the resulting distribution for the point is concentrated, with a small (bright) region.

On the bottom (b), we see the projection of a point q that is far removed from the true surface. The consequence of this is that the two imaged appearances attributed to this point are actually of different points of the real object. Since the imaged geometry for image $\boldsymbol{I}^5$ is oriented upwards, and the imaged geometry in $\boldsymbol{I}^7$ is oriented downwards, their orientation distributions are unlikely to overlap. In this case we can see that the resulting distribution exhibits no clear orientation for the point q.

Point q, whose distribution is flat and near-zero, has no true orientation, it is far removed from the surface at this stage; as the mesh evolves, it may end up somewhere quite different from its current location. p, however, has a clear target orientation, and is likely near the correct location.

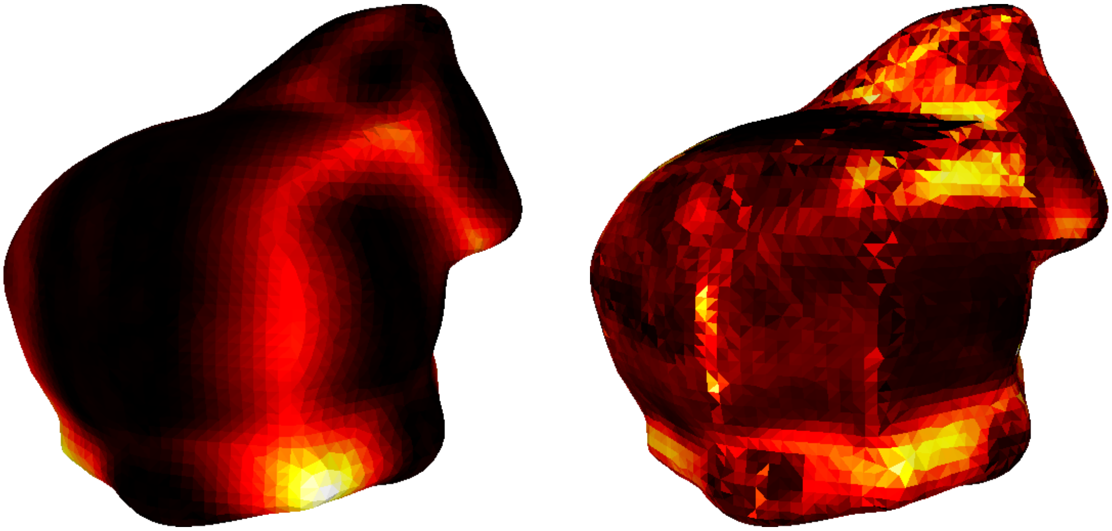
Surface patches

Now that we have seen how to unite multiple observations to derive tighter orientation distributions, we may finally turn our attention to recovering a full 3D model. By focusing on the facets of the model $f \in \boldsymbol{G}$, we provide a way to use orientation cues to deduce the full 3D geometry. The goal then is to morph the points of the mesh, so that the facet orientations are consistent with the observations.

In order to take full advantage of higher-resolution observation images, we take J uniformly distributed samples from a facet and average them (in our case $J = 6$),

$$p(\boldsymbol{I}_f^m | \boldsymbol{N}_f) = c \sum_{j=1}^{J} w_j \cdot p(\boldsymbol{I}_{f,j}^m | \boldsymbol{N}_f) \ , \tag{5.14}$$

where the weights w_j ($\sum w_j = 1$) are higher for samples near the center of the facet, and c is to ensure the distribution integrates to one. This sampling is done by division of the barycentric parametrization of the facet, and the weights are set equal to the minimal barycentric coordinate value. Here, $\boldsymbol{I}_{f,j}^m$ indicates a specific pixel—the j^{th} sample of facet f in observation m. The final likelihood for the facet is then the product of the per-image



(a) Surface to ground-truth distance **(b)** Negative log likelihood of facet orientation

Figure 5.9: Likelihood vs. actual error On the left we see the distance between the midpoint of each facet of the current mesh and the ground-truth mesh. Brighter values correspond to points further removed from the true surface. On the right we see the negative log likelihood of the current facet orientations. Brighter values correspond to less likely orientations. The strong correlation between these two images shows the accuracy of our formulation. Figure 5.8 gives the intuition behind this result.

distributions (as in Equation 5.13),

$$p(\boldsymbol{\mathcal{I}}|\boldsymbol{N}_f) = \prod_{m \in \Omega_f} p(\boldsymbol{I}_f^m|\boldsymbol{N}_f)\ . \tag{5.15}$$

Figure 5.9b shows the negative-log value of Equation 5.15 for each of the facets of an entire mesh evaluated at their current orientation. Brighter values correspond to decreased likelihood, and darker values correspond to increased likelihood. In other words, the dark regions indicate that the orientation of the facet accounts for the appearance across all of the images well. In (a) the distance of each facet to the ground-truth object. The brightest regions (near the bottom) correspond to areas between the legs that need to be carved out. Even though the likelihood is measuring the expected error in *orientation* it strongly correlated with the actual inaccuracy of the *location* of the facets. Figure 5.8 gives the intuition behind this link. The bright regions in (b) that do not appear in (a) are those where the orientation is incorrect even though the actual distance to the ground truth is quite close. This occurs in particular in the mane of the toy horse.

Probabilistic shape estimation

Now that we have described how to form the likelihood for a single facet, we may express the full likelihood of Equation 5.1, as the product over all facets,

$$p(\boldsymbol{\mathcal{I}}|\boldsymbol{G}) = \prod_{f \in \boldsymbol{G}} p(\boldsymbol{\mathcal{I}}|\boldsymbol{N}_f) \ . \tag{5.16}$$

Finally, we place three priors on the mesh itself, $p(\boldsymbol{G}) = p_c(\boldsymbol{G})p_h(\boldsymbol{G})p_e(\boldsymbol{G})$. The first prior is inspired by recent work on minimal surface constraints [26]. To propagate the shape of accurate regions to those far removed from the true surface (as in Figure 5.8b), we encourage the local curvature to be constant. This is approximated as the variance of the angles between the normal of each facet and those of the facets a certain distance away

$$p_c(\boldsymbol{G}) \propto \prod_{f \in \boldsymbol{G}} \exp\left\{\beta_c \mathrm{Var}\big[\arccos(\boldsymbol{N}_f \cdot \boldsymbol{N}_{f,n})\big]\right\} , \tag{5.17}$$

where β_c controls the strength of the prior. The set of normals $\{\boldsymbol{N}_{f,n}\}$ over which the variance is computed contains those facets that are a certain distance from the facet itself. To impose the prior only on the immediate neighborhood, for example, facets that share a single point in common with f can be used. To impose the prior more globally, facets that lie on the subsequent rings surrounding f may be used. In section 5.5.3, we describe our use of this prior.

The next prior $p_h(\boldsymbol{G})$ ensures that the mesh does not grow outside of the visual hull. It is designed to give no penalty to any point $\boldsymbol{v} \in \boldsymbol{G}$ inside the hull, while strongly penalizing points that leave the hull. Since computing point-to-mesh distances is expensive, we first detect the point on the hull $\boldsymbol{h}_v$ closest to each vertex $\boldsymbol{v}$. We also compute the surface orientation of the hull at that point $\boldsymbol{n}_v$. The (signed) distance between each vertex $\boldsymbol{v}$ of our geometry estimate and the hull can then be estimated using a simple dot product. To ensure no penalty for points within the hull, we formulate the prior with a sigmoidal potential function,

$$p_h(\boldsymbol{G}) \propto \prod_{v \in \boldsymbol{G}} \Big(1 + \exp\left\{\beta_h((\boldsymbol{v} - \boldsymbol{h}_v) \cdot \boldsymbol{n}_v)\right\}\Big)^{-1} , \tag{5.18}$$

where β_h controls the strength of the prior. If the segmentations used to make the visual hull cannot be trusted, for example, this weight can be set to zero.

The last prior $p_e(\boldsymbol{G})$ is due to our implicit assumption about the triangles that make up the mesh. It helps ensure that the triangles are roughly equilateral so that samples within each triangle may be assumed to be relatively nearby on the actual surface,

$$p_e(\boldsymbol{G}) \propto \prod_{f \in \boldsymbol{G}} \exp\left\{\beta_e \mathrm{Var}\left[f_e\right]\right\} , \tag{5.19}$$

where f_e is the length of an edge e of the facet, and again β_e controls the strength of the prior.

Parameterizing the distribution

Recall that the facet likelihoods $p(\boldsymbol{\mathcal{I}}_f|\boldsymbol{N}_f)$ are non-parametric in that they depend on the inherently non-parametric illumination environment. Because of this, a direct optimization is intractable (the visualizations in Figure 5.8 are themselves discrete approximations). In order to optimize without performing an exhaustive search, we need a way to faithfully parametrize the distribution while providing a way to avoid local minima.

To do so, we first pick a finite set of L orientations $\{\boldsymbol{N}^l\}$ by uniformly sampling the unit sphere. We then encode the distribution as a mixture of Von Mises-Fisher distributions centered at these orientations. The concentration (spread) of each distribution κ_l is proportional to the probability of the corresponding orientation $\boldsymbol{N}^l$ as computed by Equation 5.15 (in our case $\kappa_l = 200 \cdot p(\boldsymbol{N}_f)$),

$$p_{\text{approx}}(\boldsymbol{\mathcal{I}}_f|\boldsymbol{N}_f) \propto \sum_{l=1}^{L} C(\kappa_l) \exp\left\{\kappa_l \boldsymbol{N}^l \cdot \boldsymbol{N}_f\right\} \tag{5.20}$$

where $C(\kappa_l)$ is a normalization constant.

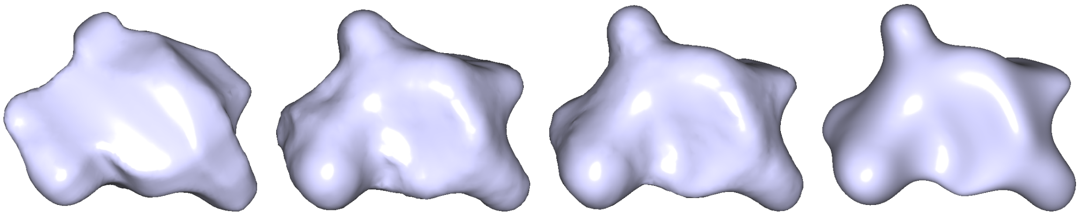
This formulation gives a continuous expression that is differentiable everywhere. The original distribution may have large areas with the same probability due to textureless regions of the illumination environment leading to ambiguous gradients. The parameterized distribution, on the other hand, will have a zero gradient only at local maxima and minima. In our case we set $L = 1024$.

5.5.2 Reflectance from shape

In order to estimate the parameters $\boldsymbol{\Psi}$ we continue with our probabilistic formulation of Equation 5.1. Here, the likelihood is the same as above, though the geometry, and hence the surface orientations of the facets $\boldsymbol{N}_f$, are kept fixed,

$$p(\boldsymbol{\mathcal{I}}|\boldsymbol{\Psi}) = \prod_{f \in \boldsymbol{G}} \prod_{m \in \Omega_f} \mathcal{N}\big(\ln(\boldsymbol{I}_f^m)|\ln(\boldsymbol{E}_f^m), \sigma^2\big) \ , \tag{5.21}$$

where Ω_f is again the set of images in which facet f appears, and $\boldsymbol{I}_f^m$ refers to the appearance at the center of the facet in image m, and $\boldsymbol{E}_f^m$ refers to its predicted irradiance.



(a) Initial (b) Mid-way (c) Final (d) True

Figure 5.10: Shape optimization iterations The geometry estimate at several stages shows how inaccurate regions are carved away as the estimate tightens around the true shape.

We, again, use the prior introduced by Lombardi and Nishino [95], which encourages realistic reflectance estimates.

$$p(\boldsymbol{\Psi}) \sim \mathcal{N}(0, \beta_\Psi \boldsymbol{\Sigma}_\Psi), \tag{5.22}$$

where the covariance $\boldsymbol{\Sigma}_\Psi$ is computed from the MERL database [97], and the scalar β_Ψ controls the prior strength.

5.5.3 Implementation and optimization

Our overall optimization scheme alternates between computing the Gaussian noise variance σ^2, and estimating the the maximum a posteriori (MAP) estimate of the reflectance parameters $\boldsymbol{\Psi}$ and then geometry $\boldsymbol{G}$. This three-step optimization framework is iterated until convergence, typically around 6 iterations. To find the MAP estimate of the reflectance parameters $\boldsymbol{\Psi}$ and geometry $\boldsymbol{G}$ we minimize the corresponding log-posteriors using gradient descent. In the case of the reflectance, this corresponds to finding reflectance coefficients $\boldsymbol{\Psi} = \{\psi_i\}$. In the case of the geometry, this corresponds to finding the 3D locations for each of the vertices of the mesh. Since the likelihood, and two of our priors are expressed in terms of the facet normals, it is important to note that these normals are themselves functions of the point locations (specifically, the normalized cross product of two facet edge vectors).

So that a single set of prior weights can be used regardless of environment settings, all images are scaled by a constant factor so that the mean intensity of the illumination environment is 1. The reflectance estimation is done using a modified version of the code released by Lombardi and Nishino [95], and the geometry estimation is implemented with the Theano library [100]. Both components run on the GPU for a combined running time of $\sim$45 min. per top-level iteration. Typically six iterations are needed to converge.

To bootstrap the process, the first step is to extract a rough estimate for the object geometry. As many other authors have done, we assume that the objects have been segmented from the background, enabling us to leverage the visual hull work of Laurentini [101] to initialize our geometry estimate. The mesh is then re-triangulated using the Poisson reconstruction [102], and small triangles are collapsed to help standardize the area of the triangles. The result of this step is shown in Figure 5.10a. With an initial geometry estimate in place, we then perform the first reflectance estimation iteration. The prior weight β_Ψ is set to 2^{-3}.

When optimizing the geometry we adopt one additional time-saving approximation. We assume that each camera is far enough away from the object that the mean viewing direction is sufficiently close to the actual per-pixel viewing direction (i.e., orthographic camera). This assumption allows us to pre-compute a single reflectance map for the camera pose that applies to every point in the image, accelerating the computation of Equation 4.6.

The curvature-based smoothing prior of Equation 5.17 is controlled based on the number of facets in the object. Since all of our meshes have approximately $10,000$ facets, we set the ring at which the smoothness is computed to be 2. For objects with more, or fewer facets the ring can be increased or decreased accordingly. The prior weights are set to $\beta_c = 2$, $\beta_h = 16$ and $\beta_e = 0.5$.

Throughout the optimization process we take into account occlusion when computing which images Ω_f contain a facet. We do not, however, model any global light transport effects such as shadows or interreflection. Additionally, samples that are observed at grazing angles (an angle greater than 75° from the viewing direction) are discarded. This threshold was chosen to avoid overly constrained likelihood distributions in the case of the geometry refinement, and misleading grazing angle reflectance properties in the case of reflectance estimation.

Finally, we note that the computation time for estimating the reflectance is greatly dependent on the size of the illumination map. In our experiments we use an illumination map of size 128×64. As we discuss in the next section, this size is sufficient for all but the most specular reflectances.

5.5.4 Experimental evaluation

We evaluate our method quantitatively on two databases: a synthetic database, and a new real-world data-set with ground-truth geometry. Since there are no other methods that recover full 3D shape with arbitrary reflectance in natural illumination, we cannot include any direct comparison.

To quantify the accuracy of our geometry estimates we compute the distance of each point on the estimated geometry to the ground-truth object. We then compute the root-mean-squared (RMS) error as a percentage of the bounding box diagonal length of the ground truth object. If, for example, the true object fits in a box with a meter diagonal, an error of 1.0% indicates a RMS error of 1cm.

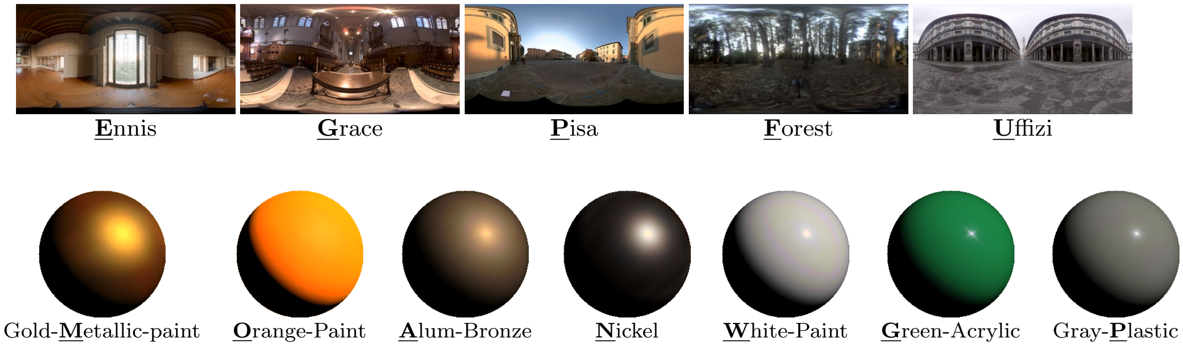


Figure 5.11: Synthetic data. Our synthetic data are formed by rendering 10 shapes [24] with 7 real-world BRDFs (bottom) [97] under 5 real-world natural illumination environments (top) [99].

5.5.5 Synthetic data evaluation

The ideal illumination environment in which to estimate the geometry of a Lambertian object is one with a few, differently colored point light sources. The resulting appearance will depend exclusively on the relative angles between the surface point orientations and the lights. Such an illumination environment would, however, provide a hopelessly sparse set of orientation cues for a mirrored object, as only a few points on the object would reflect any light at all. For a mirrored object, on the other hand, the ideal illumination environment is one in which each orientation reflects a unique illumination value. Lambertian reflectance, in this case, would hopelessly blur such detail. Real-word illuminations and reflectances, however, lie in between these extremes. In order to test the role of reflectance and illumination in shape estimation, we have performed hundreds of experiments with a wide range of real-world environments and BRDFs.

Figure 5.11 gives an overview of our synthetic data. Note the underlined letters, as they are used as indexes in Tables 5.2 and 5.1 Each of the 10 blobs from the Blobby Shapes database [24] is rendered in 5 publicly available illumination environments [99] with 7 different measured BRDFs from the MERL database [97]. When training the reflectance

		Reflectances							
		M	O	A	N	W	G	P	mean
Environments	P	0.90	0.27	0.61	0.92	0.56	0.21	0.37	0.56
	G	0.57	0.22	1.20	1.08	0.55	0.24	0.32	0.55
	F	0.67	0.26	0.75	1.19	0.50	0.20	0.32	0.50
	E	0.82	0.22	1.17	1.13	0.47	0.26	0.48	0.48
	U	0.75	0.25	1.72	0.92	0.60	0.23	0.36	0.55
	mean	0.75	0.25	1.17	1.08	0.55	0.23	0.36	0.55

Table 5.1: Synthetic results reflectance error This table is analogous to Table 5.2, except that each cell shows the average RMS reflectance error.

		Reflectances							
		M	O	A	N	W	G	P	mean
Environments	P	0.44%	0.47%	0.54%	0.57%	0.50%	0.45%	0.49%	0.49%
	G	0.47%	0.52%	0.55%	0.51%	0.68%	0.54%	0.50%	0.54%
	F	0.52%	0.49%	0.59%	0.56%	0.59%	0.57%	0.57%	0.56%
	E	0.52%	0.58%	0.54%	0.52%	0.59%	0.87%	0.67%	0.61%
	U	0.66%	0.54%	0.62%	0.66%	0.74%	0.71%	0.80%	0.68%
	mean	0.52%	0.52%	0.57%	0.56%	0.62%	0.63%	0.61%	0.57%

Table 5.2: Synthetic results geometry error Each cell shows the average RMS geometry error across the 10 blobs for an illumination (row) and reflectance (column) combination. The headers correspond to the bold letters in Fig. 5.11. For quick inspection, lower errors are given a brighter background coloring. The last row and column are means. 9 images are used in each scenario.

model and prior, the ground-truth BRDF is omitted to ensure a fair evaluation.

Table 5.2 gives an overview of our geometry and estimates when 9 images are used. Each of the rows and columns correspond to the illumination environments and reflectances shown in Figure 5.11, respectively. The last row, and column show averages for the corresponding illumination or reflectance.

The consistency of results within each column of Table 5.1 shows clearly that certain reflectances are harder to accurately estimate than others. Most notably, the two metals Alum-Bronze (A) and Nickel (N) show the highest errors. These materials exhibit some uncommon grazing angle reflectance properties that are difficult to recover. Other, reflectances such as Orange-Paint (O) and Green-Acrylic (G), however, are consistenly more accurately

		Initial	Final	Reduction
Observations	5	**2.57%**	**1.20%**	**53%**
	7	**1.67%**	**0.91%**	**46%**
	9	**1.21%**	**0.50%**	**59%**
	11	**1.06%**	**0.62%**	**41%**
	13	**0.98%**	**0.57%**	**41%**

Table 5.3: Geometry results over number of views As additional views are added both the initial error, and the final error are reduced. The overall improvement of the mesh stays somewhat constant (around 48%). Remarkably, in the case when there are 9 views, the resulting error is even lower than with 11 or 13 views. This shows that certain viewpoints are more helpful than others.

estimated.

Table 5.2 shows the geometry results. As a baseline, these numbers should be compared with the mean initial RMS error of 1.00%, so even in the worst case the error is being reduced significantly. The worst geometry estimation result, with a RMS error of 0.87%, comes from the Green-Acrylic (G) reflectance in the Ennis (E) illumination environment. This is likely due to the lack of green in the scene, making the appearance due primarily to the light coming from the doorway in the center. Due to the diverse, and smoothly varying color, intensity, and texture of the scene, the Pisa (P) illumination environment gives the best performance overall with a mean RMS of 0.45%. Only one reflectance is challenging in this environment—Nickel (N), which has only a weak diffuse component. The best reflectance, Gold-Metallic-Paint (M), has the best of both worlds—strong diffuse, and moderate specular components. This enables the appearance to capture both low-frequency and high-frequency detail of the illumination.

Table 5.3 shows the impact of additional views for a subset of our synthetic data (Blob01, Pisa, Gold Metallic Paint). In each case the views are distributed evenly around the object, that is, a new set of images is used for each case. Though it is clear that additional views improves not only the initial estimate and the final result, the viewing direction itself plays

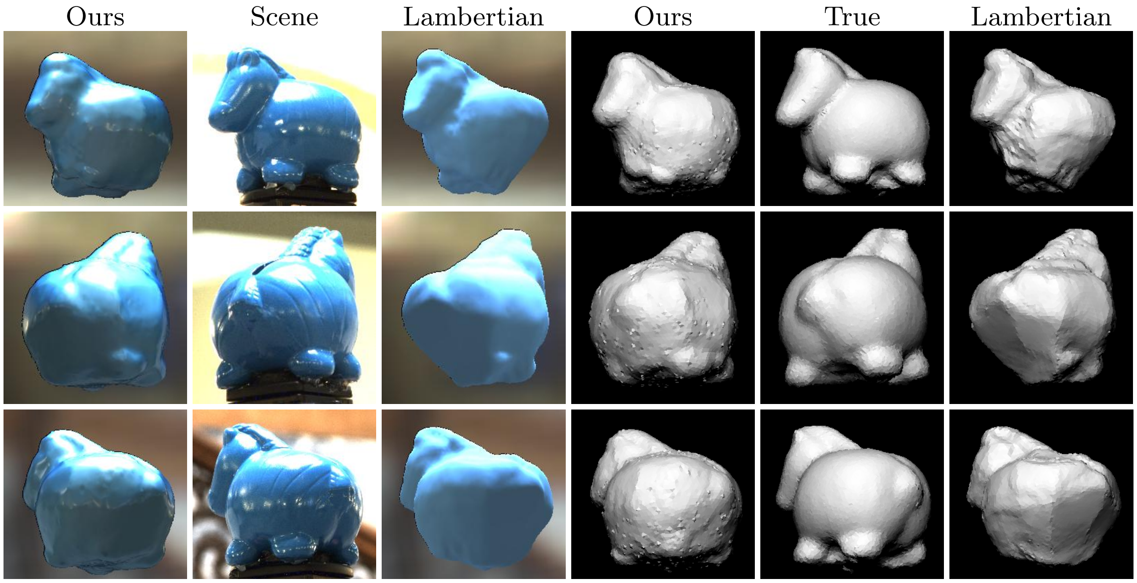


Figure 5.12: Comparison with Lambertian assumption The figure compares the results of our full pipeline with the use of the DSBRDF model (our proposed method) and assuming the object has a Lambertian reflectance. Each row corresponds to a different view point. On the left half of the image we compare our rendered result and the result when the reflectance is assumed to be Lambertian with the ground-truth appearance of the scene. On the right we show diffuse renderings to highlight the geometric differences. Highlights, which are not well explained in the Lambertian case cause the overall reflectance to become brighter. This, in turn, forces darker regions of the object to point down, towards the darkest region of the scene.

an important role. This can be seen by the lower error when 9 views are used compared to 11 or even 13 views.

5.5.6 Real-world data evaluation

To quantitatively evaluate our method on real-world objects we introduce a new data-set. The data-set contains four objects imaged in three different indoor and outdoor environments from multiple angles (approximately 18) using a tripod at two different heights. Along with the high-dynamic-range (HDR) images, the data-set contains HDR illumination maps acquired using multiple images of a steel ball, and ground-truth 3D models of the objects acquired using a laser light-stripe range finder and manually finished.

	Horse			Pig			Shell			Milk Bottle		
	Initial	Final	Delta	Initial	Final	Delta	Initial	Final	Delta	Initial	Final	Delta
Hall	1.7%	1.1%	35%	0.8%	0.7%	13%	0.8%	0.7%	13%	1.0%	0.7%	30%
Indoor	1.3%	1.0%	23%	1.1%	0.9%	18%	0.8%	0.5%	38%	1.0%	0.7%	30%
Outdoor	1.4 %	1.1%	21%	0.8%	0.5%	38%	1.2%	1.0%	17%	0.6%	0.7%	*+17%*

Table 5.4: Overview of real world results For each of our three illumination environments (rows) we evaluate each of the four objects (sections). The initial error and final error are shown, along with the relative change. The overall performance is a reduction of error by 23%.

Figure 5.12 compares one of our results to the result if a Lambertian reflectance model is used in place of the general DSBRDF model described in Section 5.4.2. Each row of the figure is an additional viewpoint. On the left we show the rendered results compared with the observed scene. On the right we show the geometry rendered diffusely to aid in inspection of the geometry. Though the object is of a material that has a strong Lambertian component, the purely Lambertian model is naturally unable to handle the wide range of intensity variation. Bright highlights increase the albedo estimate, causing the darkest regions of the image to be only explainable by a downward facing facet orientation. Consequently the back side of the toy horse, which has a dark appearance, collapses as its facets bend towards the ground.

Table 5.4 summarizes the geometry error. On average our method removes 21% of the original error. Figures 5.14, 5.15 and 5.16 show the results for all 12 of these real-world experiments. The first two columns compare an image of the scene with the corresponding final rendering of our result using the estimated geometry and reflectance. To give additional visibility into the geometry, the last thee columns show diffuse renderings of the initial, final, and true geometry. Note that the bottom side of the objects is never visible to the camera due to the support structure. As a direct consequence of this, objects with complex bases result in higher error. Note also that imaging the illumination environment necessarily

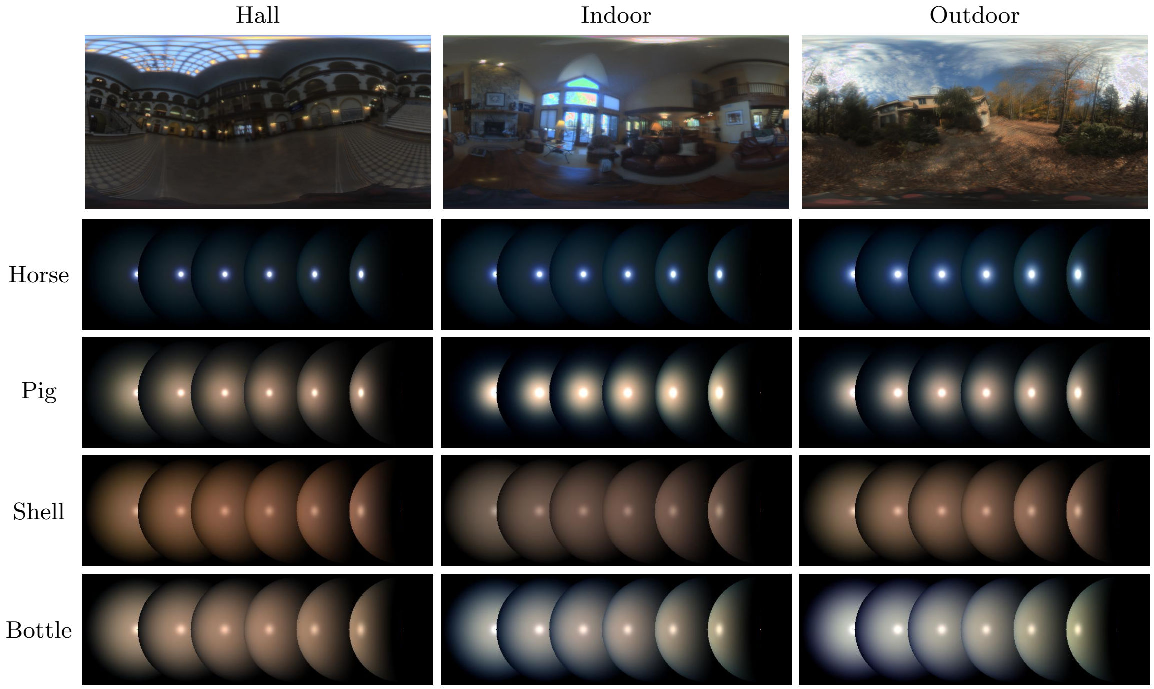


Figure 5.13: Real-world reflectance results All 12 of the estimated reflectances are shown. Each row corresponds to an object, and each column corresponds to an illumination environment. Note the overall consistency across the environments. As can be seen in the estimated reflectances, the pig object was sprayed with a diffuser after the Hall images were taken.

results in a low-pass filter of the true illumination environment as fine detail is compressed into coarse pixels. This has the impact of decreasing the sharpness of highlights in the rendered results.

Our reflectance estimates are all shown in Figure 5.13. Each row shows the results for one object, while the columns correspond to the (pictured) illumination environment. Note that the results are quite consistent across the environments. The reflectance for the milk bottle (bottom row) show some color shifting. This reflectance was consistently the hardest to estimate due to its strong specular component, and the highly planar geometry of the object. The pig object was purely specular when the hall environment images were taken. It was then sprayed with a diffuser. The change in its reflectance can be seen in the estimates as well. The purely specular reflectance was not recovered well in the hall due to inaccuracies in aligning the illumination environment, and the inherent loss of high-frequency detail. As discussed in Section 5.5.3, we use a relatively small (64×128) illumination environment in our implementation to accelerate running times. For more diffuse objects, this size is sufficient, but in the case of the purely specular object, necessary detail is lost. As a consequence the reflectance has absorbed this missing data as error. The shell and horse reflectances are quite plausible and very consistent across the different environments, as can be seen in the re-rendered results, which we discuss next.

The first illumination environment, shown in Figure 5.14, is a large hall. It has a modest amount of natural light coming from the top of the scene, but is primarily illuminated by several lights placed evenly throughout the environment. Overall we can see increased accuracy as erroneous regions are carved away. The mane of the horse is accurately recovered, as are many of the bumps on the shell. As noted above, the reflectance estimate for the pig fails to capture its purely specular nature due to the small scale of the illumination environment. Consequently the likelihood distributions remain rather flat. The result then, is that the priors have a stronger effect, and the estimated result, though a significant improvement over the initial mesh, is somewhat smoothed.

The second environment, shown in Figure 5.15, is inside a home. Though the windows provide the dominant source of light, within the home are several additional light sources. The horse is again well estimated. The primary regions of error are carved away and the shape matches the true geometry much more closely. The many peaks of the shell geometry are revealed in the result, though self-cast shadows (and some interreflection) reduce their clarity somewhat. The milk bottle shows marginal improvement, again due to its challenging shape and reflectance. With the added diffuser, our smaller scale illumination

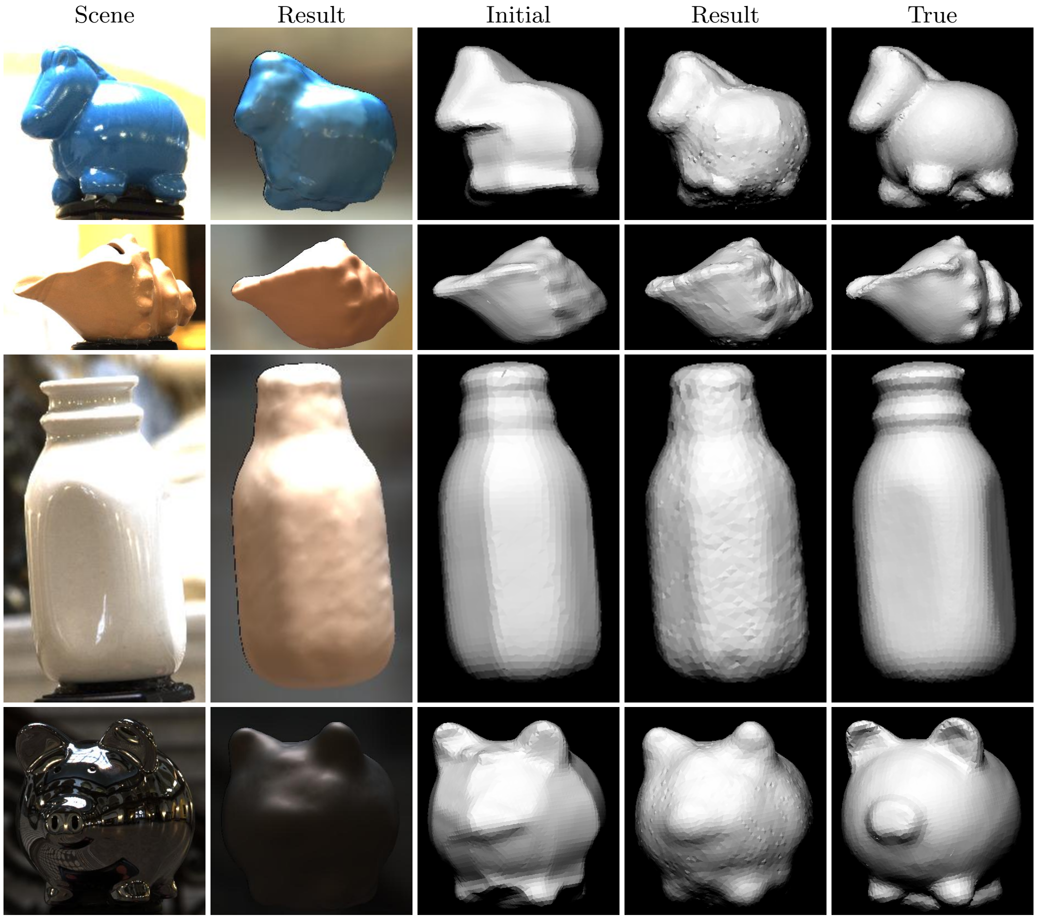


Figure 5.14: Real-world results - hall The first two columns show the captured scene and the rendered result. The next three columns compare the initial geometry, the recovered result, and the true geometry. These are rendered with a diffuse reflectance to highlight the geometric differences.

map is sufficient to estimate the reflectance of the piggy bank. Although it still frequently exhibits global illumination effects (shadows and interreflections), its 13 views give sufficient context to carve out the concave ears, and feet.

The final environment, shown in Figure 5.15, is outside a wooded home. Here again we see results consistent with the other environments. One noteworthy exception is the milk bottle. In this particular case, the initial visual hull was very close to the true object (with an error of only 0.6%). Our result, though visually more similar to the ground-truth,

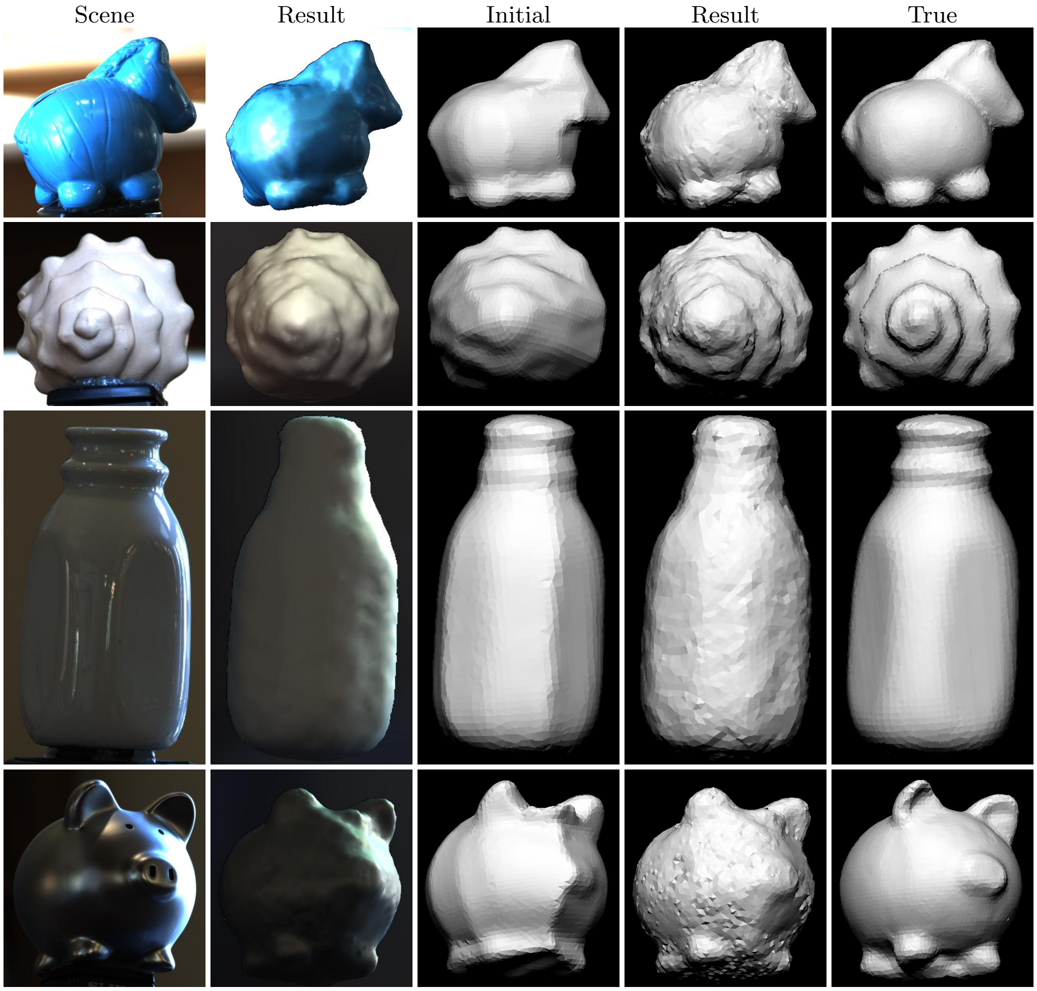


Figure 5.15: Real-world results - indoor Results for the indoor environment.

is evaluated with a slight *increase* in error (0.7%). This small amount is on the order of the inaccuracy of the ground-truth itself, as there are inherently small inaccuracies in the scanning and alignment process. Overall the results in this environment are quite strong. Note how The eyes and nose of the pig have introduced minor errors (as they are of a different material).

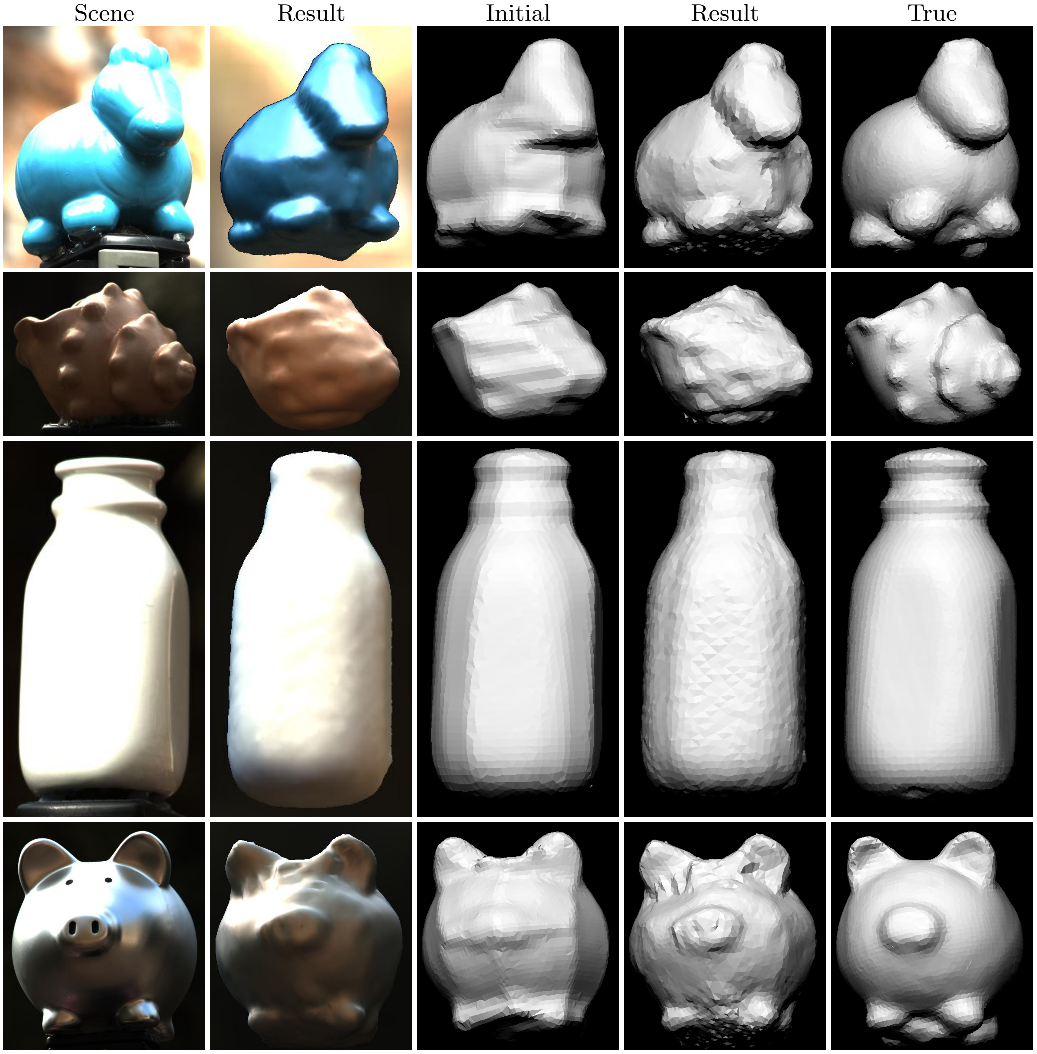


Figure 5.16: Real-world results - outdoor Results for the outdoor environment.

5.6 Summary

In this chapterwe have presented two probabilistic methods to jointly estimate the geometry and non-trivial reflectance of an object situated in complex, natural illumination. Instead of making simplifying assumptions about the illumination or reflectance we have shown how to use the complexity to our advantage. In the case were we have only a single image, our first method showed how unique scene regions, when reflected off the object, give strong clues about the local geometry, while careful priors can be used to reduce the varying degrees of ambiguity elsewhere. Our second method showed how to combine multiple observations, enabling additional views to act as further constraints on the local geometry. By fully exploiting the rich signal of real-world illumination reflected by real-world reflectance we have derived methods that recover geometry without any expensive equipment.

Chapter 6: Conclusion

The rich diversity of appearance in our world produces staggering beauty and immense challenges in computer vision. Despite the wide range, however, geometry lies at the heart of appearance. Through some process—an MRI scan, a photograph, or fracturing an object—the original geometry of an object becomes lost, leaving behind only clues. In this thesis we have investigated how to extract and exploit these clues.

6.1 Geometric deformation recovery

First we observed that intensity structures in images often capture the underlying structure of the object itself. This lead to a novel approach to image registration in which the image is treated as a 2D manifold in a 3D space. By constraining the shape of these structures as the image deforms to align with the subject, we achieved accurate registrations with physically-plausible underlying deformation fields.

In some cases more detailed geometry can be estimated from a single image. By building a more accurate 3D model of the imaged subject, the principals of this work could be used to even greater effect. For additional future work the extension to volumetric registration should be considered. In MRI imaging, for example, ensuring that the full 3D shape of constituent structures remains physically plausible during deformation will be an important step.

6.2 Fragmented geometry recovery

Next we discussed a general method for reassembling broken objects. Without making any limiting assumptions about the object the only reliable clue we have about its shape is embedded in the boundaries of the fragments. Observing that the color and geometry of adjoining fragments must match each other along and across these adjoining regions, we derived a scale-space representation of the fragment boundary geometry and photometry. This allowed us to recast the complex problem of finding matching fragments as a problem of partial image registration. As fragments are matched and aligned additional information about the alignment of each pair of fragments becomes available leading to a natural bottom-up yet globally optimal optimization framework.

The notion of human-in-the-loop reassembly has great capacity for expansion. What is challenging for humans (sifting through many fragments for potential matches, gluing pieces together) is easy when computers aid virtually. And what is challenging for computers (verifying the accuracy of matches using additional information and prior experience) is easy for humans to do. It is natural then, that this framework be expanded to give more power to each component. Future work should investigate the possibility for the two to operate in parallel to a greater extent. Through multiprocessing, the computer can be evaluating possible matches as the human considers the current configuration. The user can also provide additional insights such as noting what part of the current configuration should be filled in next. Such a seamless integration of human and computer would have a deep impact in archeology, both at a research level, and in engaging the public.

6.3 Geometric texture analysis

We then brought a fresh perspective to texture analysis. Instead of focusing on the appearance of texture, which changes with the illumination and viewing conditions, we focused instead on the underlying geometry, which remains constant. Using the fundamental properties of scale variability and self similarity, we derived two methods: one for extracting the geometry from images, and another for compactly representing the rotation-invariant scale-space of geometric texture. The results showed that our compact formulation enabled accurate texture recognition even in the face of changes in distance, and in-plane and out-of-plane rotations.

We believe these results provide a sound foundation to explore the use of geometric texture in longstanding problems in computer vision, including exploiting texture geometry to aid object recognition and scene understanding. Since many sensors exist to capture 3D geometry our insights can be used to improve these geometry estimates while still benefiting from faster acquisition. Such deployments could make rapid use of texture classification using the inherent geometry of the texture instead to provide robustness to lighting and other appearance changes.

6.4 Geometry recovery in the wild

Finally we discussed two methods for jointly estimating shape and reflectance for object with non-trivial reflectance properties in natural illumination. Though the appearance of such real-world objects exhibits great diversity, it also encodes the geometry. The orientation of each surface patch determines what hemisphere of light gets modulated and integrated to from the local appearance. By analyzing the appearance we arrived at non-parametric distributions of possible orientations. Though some appearances are inherently ambiguous, this ambiguity can be addressed with careful priors (in the case of a single image) or by uniting multiple observations to the task.

Recent work in illumination estimation gives hope that not only can shape and reflectance be estimated, but also the surrounding environment. Taken together with the successes of structure from motion future work should approach the problem of Internet scale shape estimation of objects with non-trivial reflectances. By literally filling the holes left open by past work, we will bring virtual tourism to more modern, real-world environments.

Future work should also consider a framework optimized for mobile processing. Leveraging the smart-phone ubiquity to enable end-users to scan their own objects to share with friends and family, or to 3D print replacements will be a paradigm shifting development.

6.5 Summary

Though the problems and methods presented in this thesis are diverse they share a common theme—through extracting and exploiting the latent structures embedded in observations, new approaches to challenging and longstanding problems become possible. This overall approach has enabled us both to pursue new and challenging problems, and to tackle longstanding problems in a new way. The problems presented in this thesis are only a few of those that may benefit from a careful inspection of the underlying geometry. It is our hope that future work in the wide array of computer vision problems may draw inspiration from our findings.

Bibliography

[1] M. Irani, B. Rousso, and S. Peleg. Recovery of Ego-Motion Using Image Stabilization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 454–460, 1994.

[2] G. Dedeoglu, T. Kanade, and S. Baker. The Asymmetry of Image Registration and its Application to Face Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:807–823, 2007.

[3] J. Weese, G.P. Penney, P. Desmedt, T. M. Buzug, D.L.G. Hill, and D.J. Hawkes. Voxel-Based 2-D/3-D Registration of Fluoroscopy Images and CT Scans for Image-Guided Surgery. *IEEE Trans. on Info. Technology in Biomedicine*, 1:284–293, 1997.

[4] J.B.A. Maintz and M.A. Viergever. A Survey of Medical Image Registration. *ACM Computing Surveys*, 2:1–36, 1998.

[5] K. Wang, Y. He, and H. Qin. Incorporating Rigid Structures in Non-rigid Registration Using Triangular B-Splines. *Variational, Geometric, and Level Set Methods in Computer Vision*, 3752:235, 2005.

[6] J. Kohlrausch, K. Rohr, and H.S. Stiehl. A New Class of Elastic Body Splines for Non-rigid Registration of Medical Images. *Journal of Mathematical Imaging and Vision*, 23:280, Nov. 2005.

[7] M. H. Davis, A. Khotanzad, D. P. Flamig, and S. E. Harms. Elastic Body Splines: A Physics Based Approach to Coordinate Transformation in Medical Image Matching. *IEEE Symposium on Computer-Based Medical Systems*, 8:81, 1995.

[8] J.P. Thirion. Non-rigid Matching Using Demons. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 245–251, 1996.

[9] A. Myronenko, X. Song, and M.A. Carreira-Perpinán. Free-Form Nonrigid Image Registration Using Generalized Elastic Nets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[10] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Diffeomorphic Demons: Efficient Non-parametric Image Registration. *NeuroImage*, 45:S61–72, Mar. 2009.

[11] Simon Winkelbach and Friedrich M. Wahl. Pairwise Matching of 3D Fragments Using Cluster Trees. *International Journal of Computer Vision*, 78(1):1–13, January 2008.

[12] QX Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and H. Pottmann. Reassembling Fractured Objects by Geometric Matching. *ACM Transactions on Graphics*, 25(3):569–578, July 2006.

[13] Andrew R Willis and David B Cooper. Bayesian Assembly of 3D Axially Symmetric Shapes from Fragments. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 82–89, 2004.

[14] The SHAPE Lab. Assembling Virtual Pots from 3D Measurements of Their Fragments. *Virtual Reality, Archeology, and Cultural Heritage*, pages 241–254, 2001.

[15] M Kampel and Robert Sablatnig. On 3D Mosaicing of Rotationally Symmetric Ceramic Fragments. *International Conference on Pattern Recognition*, 2:265–268, 2004.

[16] M Sağiroğlu and A Erçil. A Texture Based Approach to Reconstruction of Archaeological Finds. *Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 1–6, 2005.

[17] S. R. Fountain, T. N. Tan, and K. D. Baker. A Comparative Study of Rotation Invariant Classification and Retrieval of Texture Images. In *British Machine Vision Conference*, pages 266–275, 1998.

[18] S. Lazebnik, C. Schmid, and J. Ponce. Affine-Invariant Local Descriptors and Neighborhood Statistics for Texture Recognition. In *IEEE Int'l Conf. on Computer Vision*, 2003.

[19] T. Leung and J. Malik. Recognizing Surfaces Using Three-Dimensional Textons. In *IEEE Int'l Conf. on Computer Vision*, pages 1010–1017, 1999.

[20] M. Varma and A. Zisserman. Classifying Images of Materials: Achieving Viewpoint and Illumination Independence. In *European Conf. on Computer Vision*, pages 255–271, 2002.

[21] Robert J. Woodham. Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering*, 19(1):139–144, 1980.

[22] Shree Nayar, Katsushi Ikeuchi, and Takeo Kanade. Shape from Interreflections. *Int'l Journal of Computer Vision*, 6(3):173–195, August 1991.

[23] Neil G Alldrin and David J. Kriegman. Toward Reconstructing Surfaces With Arbitrary Isotropic Reflectance : A Stratified Photometric Stereo Approach. In *IEEE Int'l Conf. on Computer Vision*, pages 1–8, 2007.

[24] Micah K. Johnson and Edward H. Adelson. Shape Estimation in Natural Illumination. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2011.

[25] Jonathan Barron and Jitendra Malik. Color Constancy, Intrinsic Images, and Shape Estimation. In *European Conf. on Computer Vision*, 2012.

[26] Jonathan Barron and Jitendra Malik. Shape, Albedo, and Illumination from a Single Image of an Unknown Object. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 334–341, 2012.

[27] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building Rome in a day. In *IEEE Int'l Conf. on Computer Vision*, pages 72–79, September 2009.

[28] B. Zitova and J. Flusser. Image Registration Methods: A Survey. *Image and Vision Computing*, 21:977–1000, 2003.

[29] L. G. Brown. A Survey of Image Registration Techniques. *ACM Computing Surveys*, 24:325–376, 1992.

[30] B. Fischer and J. Modersitzki. Combination of Automatic Non-rigid and Landmark Based Registration: The Best of Both Worlds. *Society of Photo-Optical Instrumentation Engineers*, 5032:1037–1048, 2003.

[31] B. Fischer and J. Modersitzki. Curvature Based Image Registration. *Journal of Mathematical Imaging and Vision*, 18:81–85, 2003.

[32] S. Haker, A. Tannenbaum, and R. Kikinis. Mass Preserving Mappings and Image Registration. In *Medical Image Computing and Computer-Assisted Intervention*, pages 120–127. Springer, 2001.

[33] E. Haber and J. Modersitzki. Volume Preserving Image Registration. In *Medical Image Computing and Computer-Assisted Intervention*, pages 591–598, 2004.

[34] B. Jian and B. C. Vemuri. A Robust Algorithm for Point Set Registration Using Mixture of Gaussians. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 1246, 2005.

[35] J. Koenderink and A.J. van Doorn. Image Processing Done Right. In *European Conference on Computer Vision*, pages 158–172, 2002.

[36] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete Shells. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 62–67, 2003.

[37] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun. Discrete Quadratic Curvature Energies. *Comp. Aided Geom. Design*, 24:499–518, Nov. 2007.

[38] T. F. Coleman and Y. Li. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization*, 6:418–445, 1996.

[39] C. Cocosco, V. Kollokian, R. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online Interface to a 3D MRI Simulated Brain Database. In *Functional Mapping of the Human Brain*, volume 5, page 425, 1997.

[40] D. Rueckert, L.I. Sonoda, C. Hayes, D.L. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid Registration Using Free-Form Deformations: Application to Breast MR Images. *IEEE Transactions on Medical Imaging*, 18:712–721, Aug. 1999.

[41] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding Facial Expressions With Gabor Wavelets. In *Face and Gesture Recognition*, page 200, 1998.

[42] A. Worth. *Internet Brain Segmentation Repository*. `http://www.cma.mgh.harvard.edu/ibsr/`, 1996.

[43] Kong Weixin and B.B. Kimia. On Solving 2D and 3D Puzzles Using Curve Matching. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 583–590, 2001.

[44] Gregory C Sharp, Sang W Lee, and David K Wehe. Multiview Registration of 3D scenes by Minimizing Error Between Coordinate Frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1037–50, August 2004.

[45] K. Pulli. Multiview Registration for Large Data Sets. *International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.

[46] HJ Wolfson. On Curve Matching. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 12(5):483–489, 1990.

[47] E. Kishon, T. Hastie, and H. Wolfson. 3-D Curve Matching Using Splines. *European Conference on Computer Vision*, 427:589–591, 1990.

[48] H.C. Gama Leitao and J. Stolfi. A Multiscale Method for the Reassembly of Two-Dimensional Fragmented Objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1239–1251, September 2002.

[49] F. Mokhtarian and A. Mackworth. Scale-Based Description and Recognition of Planar Curves and Two-Dimensional Shapes. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 1(1):34–43, 1986.

[50] F. Mokhtarian. Multi-scale Description of Space Curves and Three-Dimensional Objects. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 298–303, 1988.

[51] S Suzuki and K Be. Topological Structural Analysis of Digitized Binary Images by Border Following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, April 1985.

[52] D.J. Struik. *Lectures on Classical Differential Geometry.* Addison-Wesley, Cambridge, Mass., 1950.

[53] K. Levenberg. A Method for the Solution of Certain Non-linear Problems in Least-Squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[54] Donald W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, June 1963.

[55] P.J. Besl and H.D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[56] F.S. Cohen, Z. Fan, and M.A. Patel. Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(2):192–202, February 1991.

[57] M. Smith. The analysis of surface texture using photometric stereo acquisition and gradient space domain mapping. *Image and vision computing*, 17(14):1009–1019, 1999.

[58] G. McGunnigle and M.J. Chantler. Rough Surface Classification Using Point Statistics from Photometric Stereo. *Pattern Recognition Letters*, 21:593–604, June 2000.

[59] S. Barsky and M. Petrou. Classification of 3D Rough Surfaces Using Color and Gradient Information Recovered by Color Photometric Stereo. In *SPIE Conf. on Visualization and Optimization Techniques*, 2001.

[60] A. Penirschke, M. Chantler, and M. Petrou. Illuminant Rotation Invariant Classification of 3D Surface Textures Using Lissajous's Ellipses. In *Intl. Workshop on Texture Analysis and Synthesis*, 2002.

[61] M. Chantler, M. Schmidt, M. Petrou, and G. McGunnigle. The Effect of Illuminant Rotation on Texture Filters: Lissajous's Ellipses. In *European Conf. on Computer Vision*, pages 289–303, 2002.

[62] Kirstin J. Dana and Shree K. Nayar. 3D Textured Surface Modeling. In *IEEE Workshop on the Integration of Appearance and Geometric Methods in Object Recognition*, pages 44–56, 1999.

[63] S. Barsky and M. Petrou. The 4-Source Photometric Stereo Technique For Three-Dimensional Surfaces in the Presence of Highlights and Shadows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, October 2003.

[64] Manmohan Chandraker and Sameer Agarwal. ShadowCuts: Photometric Stereo with Shadows. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[65] K. Sunkavalli, T. Zickler, and H. Pfister. Visibility Subspaces: Uncalibrated Photometric Stereo with Shadows. In *European Conf. on Computer Vision*, pages 251–264, 2010.

[66] Shree Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM Trans. on Graphics*, 25(3):935–944, July 2006.

[67] Tai-Pang Wu and Chi-Keung Tang. Photometric Stereo Via Expectation Maximization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(3):546–560, March 2010.

[68] S.J. Koppal and S.G. Narasimhan. Clustering appearance for scene analysis. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 1323–1330, 2006.

[69] J. Kim and R. Zabih. Factorial Markov Random Fields. In *European Conf. on Computer Vision*, pages 321–334, 2002.

[70] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[71] J. Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Trans. on Information Theory*, 37(1):145–151, 1991.

[72] J. Novatnack and K. Nishino. Scale-Dependent 3D Geometric Features. In *IEEE Int'l Conf. on Computer Vision*, pages 1–8, 2007.

[73] P. Bariya, J. Novatnack, G. Schwartz, and K. Nishino. 3D Geometric Scale Variability in Range Images: Features and Descriptors. *Int'l Journal of Computer Vision*, 99:232–255, September 2012.

[74] Jean-Denis Durou, Maurizio Falcone, and Manuela Sagona. Numerical Methods for Shape-from-Shading: A New Survey with Benchmarks. *Computer Vision and Image Understanding*, 109(1):22–43, January 2008.

[75] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from Shading: A Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999.

[76] Steven Seitz, James Diebel, Daniel Scharstein, Richard Szeliski, and Brian Curless. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2006.

[77] Dan B Goldman, Brian Curless, Aaron Hertzmann, and Steven M Seitz. Shape and Spatially-Varying BRDFs from Photometric Stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(6):1060–71, June 2010.

[78] Neil G Alldrin, Todd Zickler, and David J. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[79] Aaron Hertzmann and Steven M. Seitz. Example-Based Photometric Stereo: Shape Reconstruction with General, Varying BRDFs. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8):1254–64, August 2005.

[80] Rui Huang and William Smith. Shape-from-Shading Under Complex Natural Illumination. In *IEEE Int'l Conf. on Image Processing*, pages 13–16, 2011.

[81] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric Stereo with General, Unknown Lighting. *Int'l Journal of Computer Vision*, 72(3):239–257, June 2006.

[82] Yair Adato, Yuriy Vasilyev, Todd Zickler, and Ohad Ben-Shahar. Shape from specular flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(11):2054–70, November 2010.

[83] Marshall F. Tappen. Recovering Shape from a Single Image of a Mirrored Surface from Curvature Constraints. *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 2545–2552, June 2011.

[84] Steven Seitz and C. Dyer. Photorealistic Scene Reconstruction by Voxel Coloring. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997.

[85] Kiriakos Kutulakos and Steven Seitz. A Theory of Shape by Space Carving. *Int'l Journal of Computer Vision*, 38(3):199–218, 2000.

[86] George Vogiatzis, Philip Torr, and Roberto Cipolla. Multi-View Stereo via Volumetric Graph-Cuts. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 391–398, 2005.

[87] JP Pons, R Keriven, and O Faugeras. Multi-view Stereo Reconstruction and Scene Flow Estimation with a Global Image-Based Matching Score. *Int'l Journal of Computer Vision*, pages 1–24, 2007.

[88] Hailin Jin, Daniel Cremers, Anthony Yezzi, and Stefano Soatto. Shedding Light on Stereoscopic Segmentation. In *IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pages 36–42, 2004.

[89] Hailin Jin, Stefano Soatto, and Anthony Yezzi. Multi-View Stereo Reconstruction of Dense Shape and Complex Appearance. *Int'l Journal of Computer Vision*, 63(3):175–189, April 2005.

[90] Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Multiview Photometric Stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[91] Adrien Treuille, Aaron Hertzmann, and Steven Seitz. Example-Based Stereo with General BRDFs. In *European Conf. on Computer Vision*, pages 457–469, 2004.

[92] Katsushi Ikeuchi and Berthold K. P. Horn. Numerical Shape from Shading and Occluding Boundaries. *Artificial Intelligence*, 1981.

[93] K. Nishino. Directional Statistics BRDF Model. In *IEEE Int'l Conf. on Computer Vision*, pages 476–483, 2009.

[94] Ko Nishino and Stephen Lombardi. Directional Statistics-based Reflectance Model for Isotropic Bidirectional Reflectance Distribution Functions. *Journal of Optical Society America, A*, 28(1):8–18, January 2011.

[95] Stephen Lombardi and Ko Nishino. Reflectance and Naturail ILlumination from a Single Image. In *European Conf. on Computer Vision*, pages 582–595, October 2012.

[96] Szymon Rusinkiewicz. A New Change of Variables for Efficient BRDF Representation. In *Eurographics Workshop on Rendering*, pages 11–22, 1998.

[97] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Trans. on Graphics*, 22(3):759–769, July 2003.

[98] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.

[99] Debevec, Paul. Light Probe Image Gallery. *http://www.pauldebevec.com/Probes/*, 2012.

[100] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[101] Aldo Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(2), 1994.

[102] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson Surface Reconstruction. In *Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.

Vita

Geoffrey Oxholm received his Bachelor of Science in Computer Science from Lafayette College where he graduated with honors in 2004, and his Masters of Science in Computer Science from Drexel in 2010.

Publications

- "Membrane Nonrigid Image Registration" Geoffrey Oxholm and Ko Nishino. In: Proceedings of the European Conference on Computer Vision. pp 76–776. Oct., 2010
- "Aligning Surfaces Without Aligning Surfaces" Geoffrey Oxholm and Ko Nishino. In: Proceedings of the IEEE Workshop on Applications of Computer Vision. pp 174–181. Jan., 2011
- "Reassembling Thin Artifacts of Unknown Geometry" Geoffrey Oxholm and Ko Nishino. In: Proceedings of the International Symposium on Virtual Reality, Archeology and Cultural Heritage. pp 49–56. Oct., 2011
- "The Scale of Geometric Texture" Geoffrey Oxholm, Prabin Bariya, and Ko Nishino. In: Proceedings of the European Conference on Computer Vision. pp 58–71. Oct., 2012
- "Shape and Reflectance from Natural Illumination" Geoffrey Oxholm and Ko Nishino. In: Proceedings of the European Conference on Computer Vision. pp 528–541. Oct., 2012

- "A Flexible Approach to Reassembling Thin Artifacts of Unknown Geometry" Geoffrey Oxholm and Ko Nishino. In: The Journal of Cultural Heritage. pp 51–61. Jan., 2013

- "Multiview Shape and Reflectance from Natural Illumination" Geoffrey Oxholm and Ko Nishino. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Jun., 2014