REGISTRATION AND MATCHING OF LARGE GEOMETRIC DATASETS FOR CULTURAL HERITAGE APPLICATIONS

BENEDICT J. BROWN

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE

ADVISOR: SZYMON RUSINKIEWICZ

JUNE 2008

Abstract

The last decade has seen an increasing number of projects to acquire detailed 3-D representations of cultural heritage objects at museums and archaeological excavations, with a goal of improving preservation, understanding, restoration, and dissemination. However, careful study and virtual reassembly of cultural heritage objects often requires sub-millimeter precision to faithfully capture fine details, irrespective of the size and number of objects. Existing 3-D scanning technologies can produce such detail for small models with a modest amount of manual labor but do not scale to the tens of thousands of fragments that may be present at an excavation. High-precision scanners also have limited viewing volumes, making it very difficult to acquire large objects such as statues.

Most scanning technologies used in cultural heritage acquire many raw 3-D scans, each from a single viewpoint. This data does not become readily usable until the relative viewpoints of each scan have been recovered, and the data is merged into a final model. Alignment, or registration, is the process of recovering these viewpoints, and is the focus of this thesis. Assembling a large, fragmented object from its pieces involves recovering the pose of each fragment. We therefore examine the virtual reassembly problem as one of alignment.

We examine the alignment and assembly problems in cultural heritage scanning using data from the Digital Michelangelo and Theran Fresco projects. In the context of the Digital Michelangelo Project, which scanned many Michelangelo statues in Florence at approximately 0.25 mm precision, we address the challenges of aligning large, detailed range scans. Because of the statues' size, deformations due to calibration error are inevitable. We present an algorithm which accommodates warp in many large scans, thereby preserving the raw detail in the final model. We also consider the case of many small range scans, in the context of the Theran Fresco project, which is using 3-D models of fresco fragments to aid in reconstruction. Although fragments contain few range scans, they lack the detail required for stable, automatic alignment using traditional techniques. We show how to exploit the properties of fresco fragments to obtain robust, automatic alignments, and to manually correct any misalignment in only a few seconds.

Finally, we present a method for matching fresco fragments based only on geometry. Many fragments contain no decoration or distinctive edge features, so exhaustively matching edge geometry between all pairs of fragments is essential. We show how this problem relates to range scan alignment, and present a new convolution-like algorithm for efficiently computing all possible alignment of each fragment pair.

Acknowledgments

I would like to thank my advisor, Szymon Rusinkiewicz for his steady, inspirational mentorship and friendship through the seemingly endless years of graduate school. The Princeton graphics group is a very congenial place, and we graduate student regularly look to the other professors (in my case Tom Funkhouser and Adam Finkelstein) for advice and guidance well. My warm and fruitful friendships and collaborations with fellow members of the department are unforgettable: Tim Weyrich, Misha Kazhdan, Paul Calamia, Phil Shilane, Diego Nehab, Joshua Podolak, Mike Burns, Corey Toler-Franklin, any many more. I also wish to thank my thesis committee — Szymon, Tom, Adam, David Dobkin, and Ken Steiglitz — for their time and feedback.

Global, Non-Rigid Alignment is joint work with Szymon Rusinkiewicz. We also received numerous suggestions, code, and data from within the Princeton graphics group, and beyond. I would particularly like to thank Natasha Gelfand, Laurent Sabotier, the Digital Michelangelo Project at Stanford University, and the Aim@Shape repository. This project was funded in part by the Sloan Foundation and by NSF grant CCF-0347427.

The Theran Fresco Project is joint work with Corey Toler-Franklin, Diego Nehab, Michael Burns, David Dobkin, Andreas Vlachopoulos, Christos Doumas, Szymon Rusinkiewicz, and Tim Weyrich. However, many collaborators have contributed ideas, time, effort, and support to the project. I am especially grateful to our collaborators, the conservators and archaeologists at the Laboratory of Akrotiri Excavation Laboratory of Wall Paintings: Vassilis Dimitropoulos, Manolis Hamaoui, Litsa Kalambouki, Marina Papapetrou, Panagiotis Vlachos, Alexandros Zokos (conservators); Iakovos Michailidis (chief conservator); Fragoula Georma, and Niki Spanou (archaeologists). They have generously provided access, time, and cooperation in support of this project. Many of our colleagues at Princeton have also contributed invaluable effort, support and suggestions: Matt Plough, Phil Shilane, Joshua Podolak, Xiaojuan Ma, Tom Funkhouser, and the whole Tiggraph gang. Funding has come from the Kress Foundation, the Seeger Foundation, the Thera Foundation, the Cotsen Family Foundation, and NSF Grants CCF-0347427 and CCF-0702580. We especially thank Dimitri Gondicas and Peter Nomikos Jr. for his enthusiasm and help in getting the project off the ground.

Of course, graduate school is not all research. Some of my most rewarding experiences in graduate school have come from teaching. I would not have learned half as much about teaching and speaking, nor taught half so well, without the extraordinary mentorship of Bob Dondero, Donna Gabai, Andrew Appel, Doug Clark, Bob Sedgewick, and Kevin Wayne.

Finally, I would never have gotten to graduate school and beyond without the love and support of my parents Marshall and Jane, my sister Dorrit, and my grandparents. You taught me, through wisdom and example, that learning and discovering is the most exciting way to spend your life, and that if you focus on learning, everything else follows.

Contents

Abstract iii

1 Introduction 1

1.1 3-D Scanning 2

1.1.1 Time-of-Flight Scanning 3

1.1.2 Triangulation Stereo Scanning 3

1.1.3 Photometric Stereo 4

1.2 The Digital Michelangelo and *Forma Urbis* Projects 5

1.3 The Theran Fresco Project 8

1.3.1 The Destruction of Ancient Thera 9

1.3.2 Building Construction and Layout 10

1.3.3 The Wall Paintings 10

1.3.4 Excavation and Conservation of Fragments 12

1.3.5 Re-Assembling Fragments 13

1.4 Scanning Large Objects vs. Scanning Many Objects 14

1.5 Contributions 16

2 Background: Range Scan Alignment 18

2.1 Iterative Closest Points (ICP) 19

2.1.1 Stability Sampling 23

2.1.2 Distance Metrics 26
2.2 Global Rigid Alignment 27
2.3 Pairwise Non-Rigid Alignment 30
2.4 Global Non-Rigid Alignment 32

3 Global Non-Rigid Alignment 35

3.1 Global Alignment Framework 38
3.1.1 Feature Point Selection 38
3.1.2 Locally Weighted ICP for Correspondences 39
3.1.3 Global Point Positioning 41
3.1.4 Outlier Rejection 44
3.1.5 Warping with Thin-Plate Splines 45
3.2 Results 47
3.2.1 Digital Michelangelo Scanner 47
3.2.2 Other 3-D Scanners 50
3.3 Discussion 51

4 Acquisition and Alignment of Fresco Fragments 55

4.1 Design Decisions 58
4.2 Acquisition Workflow 61
4.3 Automatic Processing Pipeline 66
4.3.1 3-D Alignment 66
4.3.2 2-D to 3-D Registration 69
4.3.3 Normal Reconstruction 71

5 Background: Matching 74

5.1 Color Matching 75
5.2 Jigsaw Puzzle Matching 76

5.3 Contour Matching 77
5.4 Pottery Re-Assembly 78
5.5 Surface Matching 79
5.6 General Shape Matching 80

6 Efficient Matching of Fresco Fragments 82

6.1 Ribbon Construction 84
6.1.1 Contour Extraction 84
6.1.2 Extrusion 85
6.1.3 Choice of Extrusion Plane 86
6.2 Incremental Matching with Ribbons 87
6.2.1 Erosion Detection 89
6.2.2 Thickness Compatibility 90
6.3 Selecting Match Candidates 91
6.4 Matching Results 92
6.4.1 Wall Painting Matching Results 92
6.4.2 Forma Urbis Romae 100

7 Conclusion 101

7.1 Non-Rigid Alignment 103
7.2 Fresco Acquisition and Alignment 105
7.3 Matching Fresco Fragments 106
7.4 Future Directions in 3-D Scanning 108

Bibliography 110

List of Figures

2.1 Iterative Closest Points (ICP) 20

2.2 ICP with point-to-point and point-to-plane error metrics 21

2.3 Sequential range scan alignment 27

2.4 One-to-many non-rigid alignment 33

3.1 Rigid vs. non-rigid alignment of a *Forma Urbis Romae* fragment 35

3.2 Rigid vs. non-rigid alignments of scans from different 3-D scanners . . . 36

3.3 The global alignment framework 37

3.4 Locally weighted ICP 39

3.5 Non-rigid alignment: average feature positions are mutually inconsistent . 42

3.6 Spring-based optimization of global feature positions 43

3.7 Alignment error plots for David's head 48

3.8 Rigid vs. non-rigid alignment of David's head 52

3.9 Rigid vs. non-rigid alignment of Michelangelo's Awakening 52

3.10 Rigid vs. non-rigid alignment of a warrior statuette 53

3.11 Rigid vs. non-rigid alignment of photometric stereo scans 53

4.1 The Theran Fresco project's acquisition system 55

4.2 Color extracted from the 3-D scans and from a flatbed scanner 58

4.3 Automatic fragment acquisition with an angled scanner and turntable . . . 59

4.4 Targets for determining laser start and end parameters 63

4.5 Multi-way ICP 67

4.6 Front and back scans 67

4.7 Enhanced fragment visualization using high resolution surface normals . 72

4.8 String impressions on a wall painting fragment 72

6.1 Incremental alignment computation using ribbons 83

6.2 Original fragment edge and resampled ribbon 84

6.3 Ribbon construction pipeline 86

6.4 Dependence of the ribbon edge profile on the extraction plane 86

6.5 Example of erosion on real fragments 89

6.6 Erosion detection 90

6.7 Synthetic fresco matching results 93

6.8 Spiral fragment matches 94

6.9 White fragment matches 94

6.10 Red fragment matches 94

6.11 Precision/recall of the synthetic fresco with various matching parameters. 95

6.12 Precision/recall of the synthetic fresco for 25 mm ribbon matcher. 96

6.13 Precision/recall of the synthetic fresco for 50 mm ribbon matcher. 96

6.14 Table of synthetic fresco matches for each parameter setting 96

6.15 Precision/recall of the synthetic fresco using the best sorting method for each strip width and for ICP. 97

6.16 Precision/recall of the synthetic fresco for ICP matching. matcher. 97

6.17 Precision/recall of the restricted synthetic fresco with candidate matches selected using different error metrics. 98

6.18 *Forma Urbis Romae* matches 100

7.1 Non-rigid image alignment 104

Chapter 1

Introduction

The last decade has seen an increasing number of projects to acquire detailed 3-D representations of cultural heritage objects at museums and archaeological excavations, with a goal of improving preservation, understanding, restoration, and dissemination. However, careful study and virtual reassembly of cultural heritage objects often requires sub-millimeter precision to faithfully capture fine details, irrespective of the size and number of objects. Existing 3-D scanning technologies can produce such detail for small models with a modest amount of manual labor but do not scale to the tens of thousands of fragments that may be present at an excavation. High-precision scanners also have limited viewing volumes, making it very difficult to acquire large objects such as statues.

We examine two important bottlenecks in cultural heritage scanning using data from the Digital Michelangelo and Theran Fresco Projects: range scan alignment, and the related problem of fragment matching. The 3-D scanning technologies we consider in this thesis (and which we review below) capture shape information from a particular viewpoint; the data they acquire are called *range scans*. Multiple overlapping range scans are necessary to cover an entire object, and they must be aligned with each other and merged into a final model.

Fundamentally, range scan alignment is a matching problem: we are attempting to line up two surfaces so the shape matches. If we consider two fragments of a broken object, we could once again attempt to line up matching surfaces to detect the match. Geometry-based matching is of great importance in assembling fragmented objects from archaeological excavations. While many other cues may exist (such as color), and matching by surface alignment will be confounded by erosion, it is often the richest, most reliable cue available.

To provide a proper context for our work, we give a brief introduction to 3-D scanning technologies below. We also describe the Digital Michelangelo and *Forma Urbis Romae* projects (Section 1.2) and the Theran Fresco project (Section 1.3) to introduce the particular datasets with which this thesis is concerned, and to expose some of the particular challenges in cultural heritage scanning projects (Section 1.4).

1.1 3-D Scanning

A number of different technologies exist for acquiring 3-D models of real objects, suited to different applications. Broadly, we classify these into three categories: volume scanning (*e.g.* MRI) which acquires data about the interior as well as the surface of an object; contact scanning (*e.g.* touch probes), which measures the position of points through physical contact; and range scanners which measure the distance to the object at each pixel in a camera image. We are interested in the last type of technology, which in its different forms can acquire detailed models of any size objects without any potentially damaging physical contact. While there are many range scanning technologies, which we summarize below, a common property is that they acquire *range scans* (also called *depth images*), representing the shape of an object from a *single* point of view. A complete model requires multiple range scans taken from different viewpoints. In many cases the relative positions of these range scans is not precisely know *a priori*, leading to an alignment problem: how do you recover the precise relative positions of all range scans so they can be merged into a complete model? This thesis advances the state-of-the-art in range scan alignment to better handle certain challenging cases that arise in practice.

1.1.1 Time-of-Flight Scanning

Perhaps the simplest (conceptually) form of range scanning is time-of-flight, or LIDAR, scanning. The principle is to fire a brief pulse of laser light, and measure the time it takes to bounce off an object and return to the sensor. Given the speed of light, $c \approx 3 \times 10^8\,\mathrm{m/s}$, it takes approximately 6.7 nanoseconds for a pulse of light to travel 1 m to and object and back, and the sensor must be accurate to within 6.7 picoseconds of travel time to obtain a distance measurement with less than 1 mm error. These tolerances have the effect of limiting current time-of-flight scanners to accuracy on the order of 5 mm. On the other hand, and in contrast to the stereo technologies discussed below, this error is independent of an object's distance or size — a time-of-flight scanner can scan an elephant or a mouse with equal precision. This property makes them desirable for scanning extremely large object such as buildings, where the scanner must be placed at a considerable distance.

1.1.2 Triangulation Stereo Scanning

A second category of range scanners relies on the principle of stereo triangulation, similar to human vision. If a point is observed by a camera, its position can be localized to a ray. Using two cameras, it is localized to the intersection of two rays, which completely determines its position. Of course, each camera's projection model and their relative positions must be known. Usually these parameters are found through calibration.

Any error in localizing a point correspondence or in camera calibration results in an *angular* error in estimating a point's ray. As a consequence, the point's spatial localization error is proportional to its distance from the camera. For this reason, triangulation scanners easily achieve sub-millimeter precision for small objects (5 – 10 cm) but have difficulty maintaining accuracy as the viewing volume increases. In other words, triangulation scanners can scan a mouse with much higher precision than an elephant. In practice, as we will see in Chapter 3, it is possible to mitigate this error using non-rigid alignment techniques.

Our discussion of triangulation so far has assumed we have corresponding points from two cameras. In fact, finding these correspondences is a significant challenge. The approaches taken to this problem naturally subdivide triangulation methods in *passive* methods, which use only information inherent to the object or scene, and *active* methods, which project some form of light to aid in correspondence finding. *Structured light* scanners project a high-frequency pattern of light into the scene to create texture, making it easier to find correspondences. *Laser triangulation* scanners replace one camera with a laser stripe generator. Because the stripe generator is effectively sending out rays of light at a known angle, the remaining camera need only observe them in the scene to triangulate a position.

1.1.3 Photometric Stereo

The third major 3-D scanning technology, photometric stereo, does not measure the depth of points *per se*. Rather, it measures the *orientation* of each point, *i.e.* the surface derivative. Orientation is directly useful in many visualization applications, and photometric stereo can more precisely resolve fine details than triangulation stereo.

To understand how photometric stereo works, it is necessary to introduce some background on surface reflectance models. Suppose a surface point **p** has normal direction **n**. When light hits **p**, it is scattered in all directions, but not necessarily in equal proportions. For example, on a mirrored surface, all light will bounce in the mirror, or specular, direction. However, if the surface is diffuse and **p** is irradiated from a specific direction **l** (as is the case when the light source is a small point), the amount of light scattered in any given direction is proportional to $\mathbf{n} \cdot \mathbf{l}$. Note that in this case, the amount of light scattered is indeed equal in all direction. By sequential lighting the scene from (at least) three known point lights, enough constraints are obtained to determine both the surface orientation and its albedo.

Note that computing the irradiance direction from a point light requires knowledge of both the light *and* surface locations. It may seem paradoxical to recover surface normals from photometric stereo when we already have geometry, however photometric stereo can resolve much finer details than triangulation stereo. The precision of photometric normals often exceeds that of normals computed from geometry by an order of magnitude. Furthermore, in many cases proxy geometry can be used to measure normals; for relatively flat objects, a plane suffices.

1.2 The Digital Michelangelo and *Forma Urbis* Projects

The Digital Michelangelo project [Levoy *et al.* 2000] is one of the earliest large-scale 3-D scanning projects in the cultural heritage domain. Over the course of a year, a team of researchers undertook to scan many of Michelangelo's statues in Florence. In order to capture detailed chisel marks, both for analysis and to accurately render self-shadowing effects (which Michelangelo used to heighten the sense of perspective), it was necessary to acquire geometry at roughly 0.25 mm resolution. This resolution is most easily achieved with a laser triangulation scanner whose error is proportional to the viewing volume. Working against this was the size of the statues: Michelangelo's David stands 7.5 m tall on its pedestal.

In additional to the stringent precision requirements, cultural heritage projects present a variety of practical constraints, from which the Digital Michelangelo project did not escape. Statues cannot be moved, and must be scanned *in situ*, so the scanner had to be transportable. For safety reasons, it also had to stay a certain distance from the statues at all times. In order to "see" as much of each statue's surface as possible, it was therefore essential that the scan head be mountable in different configurations on the scanning gantry. (Even so, certain areas were impossible to scan, such as David's armpit.)

Since the error in triangulation scanners is proportional to the viewing volume, the combination of high resolution and large object size imposes very tight calibration constraints. Here we interpret calibration in a broad sense, including not only the projection parameters of the camera and laser position, but also the scan head's position on the gantry and the stability of the gantry itself. Each of these practical constraints forces tradeoffs in the scanner design that make it harder to calibrate precisely. In turn, errors in any of the calibration parameters introduce bias into the measurements which manifests as a low-frequency warp in the scans. While a different scan head design could have reduced the warp, our tests of smaller, commercial 3-D scanners indicate it is probably not possible to eliminate it completely (Figure 3.2). In the best of cases, it may be possible to model the warp mathematically (although describing it completely is equivalent to calibrating the scanner!), but in practice it has proven too complex. Furthermore, in some cases, such as a slightly unstable gantry, the “miscalibration” is not an analytic function that can be precisely accounted for.

Unlike measurement noise, the warp we observe in Digital Michelangelo data does not average out across multiple overlapping range scans. Instead, as we discuss in Chapter 3, it results in misalignment of scans, blurring out the very detail the project set out to capture. To overcome this, we present an alignment algorithm that accounts for the warp. In this way, we contain the error within the low frequency, rather than allowing it to creep into high frequency detail via the alignment process.

In conjunction with the Digital Michelangelo project, the same team undertook to scan all extant fragments of the *Forma Urbis Romae* [Koller *et al.* 2006], a marble map of Rome from the early 3rd century A.D. The map is a blueprint of Rome, including not only buildings, but rooms, doorways, and even staircases, all of them marked by incisions less than 1 mm deep. Approximately 10% of the map’s surface is presently known, and is broken into 1273 fragments of varying size and thickness. Efforts have been underway for nearly 500 years to assemble these fragments, and thereby learn as much as possible about ancient Rome. However, progress has been slow because there are many gaps, fragment edges are often highly eroded, and not all fragments contain incisions.

Using computer models, new methods become available to search for matches. For example, it is possible to extend the incisions associated with roads and buildings to find matches between non-adjoining fragments [Koller and Levoy 2005]. The computer can also easily account for constraints such as marble veining direction, fragment thickness, and location and orientation of mounting brackets. Since the temple wall on which the map was mounted still exists, many of these bracket locations are known, and can be correlated with features on the fragment backs. Finally, computers can consider matches based on the edge geometry of fragments, which is completely impossible using the 2-D puzzle-piece replicas that people have previously used when searching for matches. For unmarked fragments, edge geometry may be the only means of recovering matches.

However, the *Forma Urbis Romae* dataset exhibits technical challenges similar to both the Digital Michelangelo scans, and the Theran Fresco data we present below. On the one hand, the large fragments were scanned with the Digital Michelangelo scanner, and exhibit the expected calibration and alignment problems. These are exacerbated by the flatness and paucity of detail on fragments, which conspire to make alignment under-constrained and unstable, as with fresco fragments. Fragment matching, too, is a similar problem in the *Forma Urbis* and Theran Fresco projects. Although the matching algorithm we present in Chapter 6 is geared to the Theran Fresco fragments (which are smaller, thinner, and less eroded), we show that it can also be applied essentially unmodified to *Forma Urbis* fragments.

1.3 The Theran Fresco Project

Much of the work presented in this thesis is part of the Theran Fresco Project, a collaboration between the Akrotiri Excavation, Thera and Princeton University, and concerns technical aspects of the acquisition, alignment and matching systems we are designing to aid in reconstruction of frescoes at the site. We give a short introduction to the excavation, conservation, and archaeological contexts here, drawing on the more complete descriptions *The Wall-Paintings of Thera* [Doumas 1992] and *Santorini: A Guide to the Island and its Archaeological Treasures* [Doumas 1999]. The site of Akrotiri is at the Southern tip of the volcanic Greek island of Santorini, also known as Thera, which is itself the Southernmost island in the Cyclades. Excavations began in 1967, led by Prof. Spyridon Marinatos in 1967 under the aegis of the Greek Archaeological Society, and have been overseen since his death in 1974 by Prof. Christos Doumas. While the Late Bronze Age town is the last and most complete pre-eruption settlement, pottery has been found dating as early as the third millennium B.C., showing continual settlement since at least that time.

1.3.1 The Destruction of Ancient Thera

The present island is the crescent-shaped edge of a volcanic crater, the remainder of the island having been destroyed in a devastating explosion in approximately 1630 B.C. (according to current estimates). Prior to this time, a Cycladic civilization, with substantial Minoan influence in its later period, flourished on the island. The Akrotiri excavations cover a portion of their port city and, as with Pompeii, have been extremely well preserved by the volcanic ash. They consequently give the most comprehensive picture of a Late Bronze Age civilization available. Of particular importance are the wall paintings, which are extensive, well-preserved, and remarkably complete (that is, most of the material for many wall paintings is extant). At no other site are such extensive remains of contemporary wall paintings found, making them unique. Our collaboration concerns the use of computer graphics techniques to aid in their reconstruction.

The town was destroyed twice by earthquakes prior to the volcano's explosion. The first destruction occurred long before, and allowed time for substantial rebuilding. The excavations have therefore revealed a mix of Middle and Late Cycladic pottery in the fill from the first destruction. Wall painting fragments pre-dating this destruction have been found in both the fill and building foundations, however the complete wall paintings are Late Cycladic, and prove there was time to rebuild before the second round of earthquakes and eruption. The second round of destruction must have started with relatively minor earthquakes, giving the inhabitants time and warning to evacuate, as no bodies and few small valuables have been found to date. These light tremors were followed by larger ones which once again destroyed the town, and finally a series of eruptions culminated in the islands destruction.

1.3.2 Building Construction and Layout

The Late Bronze Age town revealed by the excavations (or more precisely the portion of town revealed) contains large, two- and three-story buildings, with narrow streets running between them. The buildings were largely constructed out of mud bricks and stones, but contained wooden reinforcements as well. Hewn (ashlar) stone blocks were also used sometimes to frame window and doors and at the corners of buildings. Especially opulent buildings had complete ashlar facades (*e.g.* Xesté 3), or were even of complete ashlar construction (*e.g.* Xesté 4). Buildings were connected to a sewer system constructed below the street via pipes embedded in the walls.

Typically, the ground floor was dedicated to workshops, storage, and a mill. Living quarters were located on the upper floors, and their walls were plastered with lime plaster (as opposed to clay), and many walls were painted. These extensive wall paintings are the most important find at Akrotiri, because of their unique state of preservation.

1.3.3 The Wall Paintings

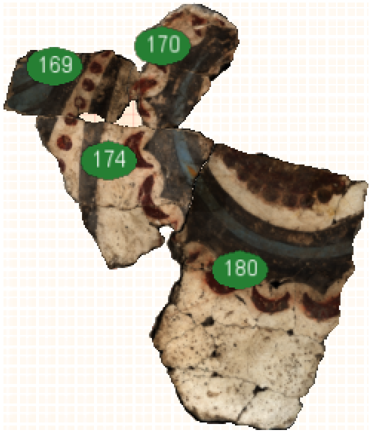
The Theran wall paintings are early examples of frescoes — paintings on plaster. Just as in Crete, the Therans prepared the wall with a layer of mud, followed by a layer of coarse plaster (1.5 cm thick), followed by on or more layers of fine plaster (.5 cm thick). The plaster was rubbed smooth using pebbles from the sea.

In *buon fresco* technique, pigments are directly applied to, and absorbed by, wet lime plaster, forming a very durable image. *Fresco secco* is paint applied to dry plaster; the color remains only on the surface and is therefore prone to flaking off. The Theran artists generally started their wall paintings on wet plaster, but do not seem to have taken care to keep the plaster wet. The wall paintings are therefore a mix of well-preserved *buon fresco*, and of *fresco secco*. Theran artists had a relatively limited set of pigments with which to paint. The principal colors used were red, yellow, black, blue, and white (the lime plaster itself). All the principal pigments are of mineral origin, as lime plaster tends to corrode organic material very quickly. Pigments were mixed, most often to produce grey, brown, pink.

Because the time to paint frescoes is (at least nominally) limited by the time it takes the plaster to dry, there is little time to prepare a careful underdrawing. Instead, as in Crete, bands of color or decoration were delineated by string impressions. More complex designs were mapped out either with string impressions or narrow incisions. Finally, in some cases outlines were sketched lightly. However, this work had to occur quickly, and the outlines were not slavishly followed. Therefore, while string impressions and incisions are important cues not only for matching fragments, but for inferring decoration which is not preserved, it provides only approximate information.

String Impressions

Uniquely, much of the original plaster material remains at Akrotiri; from the tens of thousands of small lime-plaster fragments uncovered at the site (typically less than 10 cm across and less than 1 cm thick), it is possible to largely reconstruct interior walls of buildings and thereby recover important clues to the culture, technology, and architecture of ancient Thera. The fragments tested in this paper come from a huge wall painting (approximately



Fragments from the spiral motif

5.20 m × 3.20 m) depicting symmetrical pairs of spirals (Figure 1.3.3); the original wall was not preserved at all. Thus, the restoration of this composition is also important to studying the architecture and function of the collapsed third floor of the building [Vlachopoulos 2008, 454, Figure 41.47–50].

1.3.4 Excavation and Conservation of Fragments

Although durable compared to many painting techniques, frescoes (and plaster generally) nevertheless becomes fragile after thousands of years. Among other problems, limestone undergoes a chemical change when it comes in contact with water, so fragments must be protected from any moisture. (The reaction reverses itself during drying, and is fundamental to the production of lime plaster.) Each fragment must also be cleaned and stabilized with glue to prevent erosion.

When a collection of plaster fragments is found, excavation work is halted while a conservator stabilizes them for removal. Fragments are brushed off, and moisture is eliminated by impregnating the plaster with acetone. The fragment is stabilized using increasing concentrations of *movilit* glue dissolved in acetone. Fragments must dry completely between each application, so that the entire process may take multiple days depending on the ambient humidity. Finally, gauze is attached to the fragments, again using *movilit* in acetone, and they are removed to the lab where for further conservation and assembly. This process stabilizes fragments enough to support a certain amount of handling. So while, minimizing handling is desirable, non-contact 3-D scanning is safe and practical. Additionally, it often preserves fracture surfaces well enough that a “lock” can be felt between matching fragments, from matching shape details at the sub-millimeter level.

In some cases, entire sections of wall are found intact, and can be removed as a single piece. Other sections are broken into fragment which nevertheless remain together. These can be backed by a single piece of gauze, and removed "pre-assembled." However, most fragments come individually, and must be assembled by conservators in the lab. Matches may be recorded on the fragments themselves using a removable pigment, until the fragments are glued together. When entire walls are completed, they are mounted in an aluminum frame, with spaces filled in using a distinctive-colored plaster. A final cleaning is performed on the assembled fresco to remove any residual dirt or glue.

An essential aspect of the conservation process is that it be *reversible*. No step should be undertaken which cannot be undone should it prove harmful in the long term, or should better methods be developed in the future. The glue and plaster used to fill spaces can both be completely removed without affecting the original fragments, as can the match markings. Even after a section of wall is mounted in a frame, it can be removed, or sections of fill plaster can be replaced with additional original fragments. Scrupulously ensuring reversability guarantees that steps taken now will not hinder new restoration or study in the future, and is a central tenet of modern conservation.

1.3.5 Re-Assembling Fragments

In fact, the majority of the conservators' time is in finding matches between the thousands of fragments. While the conservators use considerable skill and experience, along with cues such as the decoration, fragment thickness, fragment backs (which may contain features such as reed impressions), and location of finds, the sheer number of fragments makes the task inordinately difficult. That many fragments are a solid color only increases the challenge. Many matches are found by collecting many candidates (based on color, thickness, find location, etc.) to match a specific fragment location, and trying each one to see if they “lock” into place. If a possible match is found, the conservator may gently clean the edges of each fragment to clear away any remaining particles of dirt, try the match again, and repeat the process until the match is solid. This is a slow and painstaking process that can occupy up to 75% of a conservator’s time.

The second half of this thesis is concerned with the application of computers to this matching problem. Although computers cannot perform any of the conservation or assembly work, they may be able to sift through possible matches very rapidly, reducing the number of candidate matches a person must test. Based on results using a test fresco and limited datasets from the excavations, we believe we can produce enough useful candidates to compensate for the time spent acquiring computer models of each fragment. At the same time, computer models provide new tools for the visualization and study of the wall paintings, and reduce the amount of physical handling required.

It might seem less important to assemble all white and solid-color fragments than it is to assemble decorated areas, since they convey no pictorial information. However, beyond completeness, assembling “blank” areas of wall allows pictures to be positioned relative to each other. It also provides important information about the building architecture. Between the earthquakes and eruption, buildings suffered considerable damage, and especially upper floor were often reduced to rubble. Reconstructing wall paintings, including undecorated areas, can provide the dimensions of rooms, as well as the locations and size of doors, windows, and staircases.

1.4 Scanning Large Objects vs. Scanning Many Objects

The Digital Michelangelo and Theran Fresco projects expose two aspects of large scale cultural heritage projects. The Digital Michelangelo Project acquired only 10 statues, but each one is extremely large relative to the nominal viewing volume of a triangulation scanner. The demand for 0.25 mm resolution on statues as tall as 5 m, pushed (and continues to push) the limit of scanning technologies. The tradeoff is thousands of scans per statue which must be simultaneously aligned to each other, and a non-rigid warp in each scan due to scanner calibration error. Range scans from this project extend up to 50 cm across, and while the detail is accurate, they typically “curl” up to 1–2 mm (more in extreme cases) from end to end. In contrast, as we show in Chapter 3, commercial off-the-shelf laser triangulation scanners with a nominal viewing volume 5–10 cm across exhibit a warp of 0.1–0.2 mm. In other words, the warp present in Digital Michelangelo scans appears to stem as much from inherent limits in the ability to calibrate mechanical components as from design decisions in the particular scanner used.

A statue such as Michelangelo’s David takes days or weeks to scan, with extensive supervision throughout. In this context, additional supervision of the scan alignment process is reasonable: much of it can occur in parallel with the scanning, and the overall manual effort related to alignment is balanced with respect to scanning effort. The main technical challenges are therefore constructing a high-resolution scanner with a large enough viewing volume, and compensating for the inevitable warp in the range scans.

On the other hand, the Theran Fresco project aims to acquire tens of thousands of small fresco fragments, each of which can be completely scanned on commodity hardware in minutes. The supervision consists of placing a fragment on a turntable, clicking a button, turning the fragment over when it has been completely scanned so that both the front and back surfaces are acquired, clicking another button, and removing the fragment from the turntable. Using the system described in Chapter 4, a single user can operate four scanners in parallel to acquire 20 fragments per

Scanning a fragment

hour. Even at this speed, it would take an operator two months to scan 10,000 fragments, working eight hours per day. Aligning the data using traditional alignment techniques would more than double the effort required, and, as we show in Section 4.3.1, would frequently fail due the lack of sufficient geometric detail on many fragments.

The challenge of scanning many small objects lies therefore in robust, *automatic* alignment techniques that minimize human effort. Although computation time is an issue, it is less important: computers are much cheaper than people, and the disparity is constantly increasing. We present methods for robustly and automatically aligning range scans taken on a turntable in Section 4.1, and for aligning two sequences of turntable scans to each other (for objects with a flat front surface) in Section 4.3.

1.5 Contributions

The remainder of this thesis is organized around the dual problems of alignment and matching. We review related alignment work in Chapter 2, before discussing our alignment contributions in Chapters 3 and 4. We then summarize the matching literature in Chapter 5, and present our alignment-inspired matching algorithm in Chapter 6. Taking a unified view of the two problems, we makes the following contributions:

- A new non-rigid alignment algorithm that handles arbitrarily many range scans, and accommodates some warp in the data (Chapter 3). Using this algorithm, we show significant improvement in alignments of models from the Digital Michelangelo and *Forma Urbis Romae* projects, leading to improved detail in the final models [Brown and Rusinkiewicz 2007].

- A system for efficiently acquiring shape, color, and normals (surface orientation information) from thousands of fresco fragments (Chapter 4). In order to minimize manual effort, essential components of this system are robust, automatic methods for aligning range scans to each other, for registering color and normals to the 3-D models, and a simple interface to verify models and correct errors. As part of the Theran Fresco project, we have deployed this system at the Akrotiri Excavations [Brown *et al.* 2008].

- An efficient algorithm for matching fresco fragments based on detailed shape information on the fragment edges (Chapter 6). Starting from standard alignment techniques, we show how to make an exhaustive search for matches tractable using convolution-like optimizations [Brown *et al.* 2008].

Chapter 2

Background: Range Scan Alignment

To start our discussion of alignment, we review the Iterative Closest Points (ICP) algorithm (Section 2.1), paying particular attention to the relatively recent analyses of geometric stability and different distance metrics (Sections 2.1.1 and 2.1.2). ICP is a fundamental part of most alignment algorithms, including the one we present in Chapter 3, and is the prime example of a *pairwise rigid* technique (as opposed to a *global* and/or *non-rigid* technique). Rigid vs. nonrigid refers to the transformation used to align a pair of scans: rigid techniques allow only a rigid-body (rotation and translation) transformation, while nonrigid algorithms permit more flexible alignments. Local algorithms consider only a pair of range scans, whereas global algorithms consider the alignment of all scans in a larger set to each other, which is important to ensure that alignment error is evenly distributed across all scans. In this chapter, we will also review previously existing global rigid (Section 2.2), and pairwise and global nonrigid algorithms (Sections 2.3 and 2.4).

A priori, range scan alignment should fall into the global, rigid-body category. Assuming the object being scanned does not deform, it should be possible to align all scans using rigid-body transforms. Because there are almost always more than two scans per object, a global alignment is required. However, all standard global alignment algorithms work by optimizing over many pairwise alignments, so most of our discussion will focus on the pairwise case. Ultimately, we must address the problem of warped scans, leading to the nonrigid algorithm presented in Chapter 3.

The idea of nonrigid alignment is not new in this thesis. Nonrigid alignment problems arise most frequently in contexts where the object deforms during scanning, or when aligning scans of different objects. A common example of this problem arises in medical imaging, where it is useful to align an MRI of a patient's organ (*e.g.* his or her brain) to a template model in which all structures are labeled, or to a scan taken during a previous medical examination. Another context involves aligning articulated objects, such as people, in different poses or to templates. Still other applications wish to scan a person in motion. We will discuss the various nonrigid alignment techniques that have been developed for these problems, and their relationship to range scan alignment.

2.1 Iterative Closest Points (ICP)

At the heart of most range scan alignment algorithms lies iterative closest points, or ICP [Besl and McKay 1992; Chen and Medioni 1992]. ICP is a pairwise rigid algorithm, but also shows up in the inner loop of most global algorithms. The idea is simple: assume two point sets (scans) A and B are approximately aligned; then for each point in A, assume the closest point in B corresponds to it. (Closest points are found using a k-D tree.) Using the point correspondences found, compute a new rigid-body alignment of A and B (several efficient, closed-form methods exist to find the best rigid-body alignment of two point sets in the least-squares sense [Eggert *et al.* 1997]). Repeat the process until the alignment converges (Figure 2.1). In practice, to improve performance, a random subset of points are used, typically not exceeding a few hundred. A new set of points is chosen at each iteration.

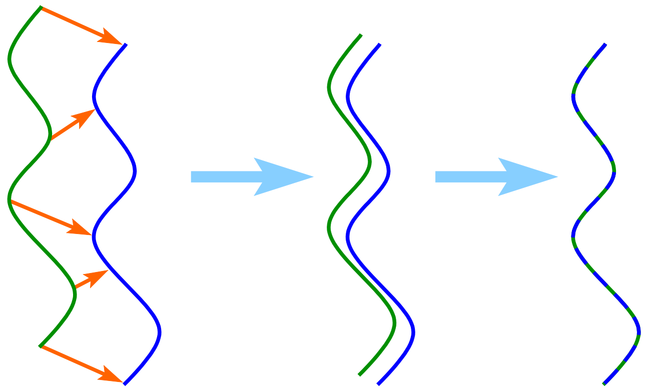


Figure 2.1: Iterative Closest Points (ICP) aligns two scans by alternately computing closest point pairs on two scans and recomputing the best rigid-body alignment.

In the above formulation, ICP is guaranteed to converge to a local minimum (assuming that the point set used does not change between iterations). Recomputing the best rigid-body alignment minimizes the sum (or mean) squared distance between point pairs and recomputing closest points is also nonincreasing. Therefore, each stage of the iteration must decrease the error, which guarantees convergence. Note that ICP converges to a *local* minimum, which is not necessarily the correct alignment. Note that, because randomly sampling points at each iteration, while useful in practice, it breaks the proof of convergence. Several standard modifications to the algorithm, described below, significantly improve ICP's likelihood of finding the correct alignment, while reducing the number of iterations required.

Non-Overlap Detection Range scans do not generally overlap completely, since the purpose is to cover many views of an object. However if a point p on scan A does not overlap scan B, there is no correct correspondence to p on B. The closest point on B will *always* be an incorrect correspondence, and will pull ICP toward an incorrect alignment. Similarly, a very poor correspondence to p will be counterproductive. To address these issues, implementations use a maximum distance threshold to the nearest neighbor search; if there is no point on B closer to p than this threshold, no correspondence is recorded. (This heuristic always bounds the k-D tree lookup time, improving performance.) A second heuristic marks all point on the boundary of A and B. Any correspondence including on of these points is discarded, as non-overlapping points on A will likely be closest to boundary points on B and vice versa.

Point-to-Plane Error Metric Besl and McKay [1992] search for closest points using the standard Euclidean distance metric. This is usually referred to as *point-to-point* ICP. When surfaces are relatively flat, and misaligned along the plane's tangent, convergence will be slow. As shown in Figure 2.2, a point p located on a bump on the scan will map to its correct correspondence, but all other points will have incorrect correspondences with low error. The result is that one point correspondence will vote for the correct alignment, while all others vote for the status quo, and the alignment will improve very slowly.

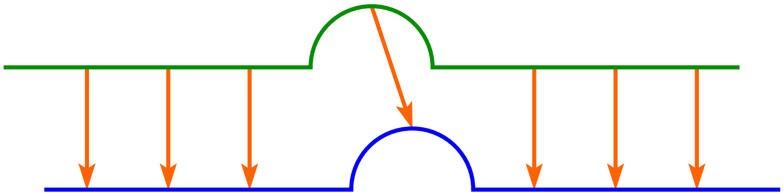


Figure 2.2: Using a point-to-point error metric, the good correspondences along the flat surface will overwhelm the correspondence on the bump, resulting in slow convergence. With a point-to-plane metric, the flat regions can slide along each other without increasing error.

An alternative, proposed by Chen and Medioni [1992], measures the distance between two oriented points p and q as the distance of p to the plane defined by q and its normal (a *point-to-plane* metric). Under this metric, planar regions on the two scans can slide arbitrarily far with respect to each other in a single iteration. The point-to-plane metric has proven experimentally to require fewer iterations to converge, and to be more robust to local minima, as long as the alignment between the two surfaces is already close. We discuss the theoretical basis for this behavior in Section 2.1.2.

Finding the closest point under the point-to-plane metric is more difficult than under point-to-point. In practice, a common approximation is to look up the closest point in the Euclidean sense in a *k*-D tree, but compute the new rigid-body alignment to minimize the point-to-plane error between the resulting point sets. While this calculation is not closed-form, it can be efficiently approximated by linearizing the rotation, assuming the new transformation is close to the current one [Gelfand *et al.* 2003].

Compatibility Constraints Another standard improvement to ICP is replacing the closest point search with a closest *compatible* point search. The most common compatibility constraint is that the two points should be oriented in similar directions (which is easily implemented as a threshold on the dot product between the two normals) [Rusinkiewicz and Levoy 2001]. Conceptually, this is an approximation of a closest point search in the six-dimensional space of position and orientation that avoids the degradation in efficiency of high-dimensional *k*-D trees. We have also found it useful to enforce a similar mean curvature constraint on low-noise data (in particular for the Digital Michelangelo and *Forma Urbis Romae* projects) [Brown and Rusinkiewicz 2007]. Because curvature is a second derivative, it is more susceptible to noise than normal computation, making it less useful on noisy data. (To reduce sensitivity to noise, we compute curvature on smoothed normals, then smooth the curvature values. For Digital Michelangelo models, the standard deviation for normal smoothing is twice the median edge length; for curvature smoothing it is the median edge length. *Forma Urbis* fragments have sharper changes in normal direction, so we smooth normals by the median edge length to avoid obliterating them.) For range scans that contain color information, this also provides a useful compatibility constraint. Sharp *et al.* [2002] uses both principal curvatures and higher-order features (moments and spherical harmonic descriptors) to augment the distance metric and increase discriminability. In that work, closest points are found in a high-dimensional space which includes both position and feature values, rather than using features as a compatibility constraint.

Outlier Rejection Among point correspondences that pass all the previous tests, some will of course be more accurate than others. Effective outlier rejection to prune poor correspondences will further improve the robustness and efficiency of ICP. One technique is to retain a fixed percentage of correspondences, keeping those which are closest together. Alternatively, it is possible to keep only correspondences whose distance is within a fixed multiple of the median correspondence distance.

2.1.1 Stability Sampling

The standard ICP algorithm uniformly samples points from a mesh to perform alignment. While uniform sampling is computationally efficient, it does not necessarily provide the fastest possible convergence. To illustrate its shortcoming, consider again the example of a flat surface with a small bump (Figure 2.2). The flat surface does not contain enough information to constrain the alignment — a translational degree of freedom remains. It is therefore essential to select at least one point on the bump at each ICP iteration in order to ensure convergence. With uniform sampling, it is likely that all samples will be chosen from the flat surface.

By asking the question, "How much will a small change in the alignment transformation affect our error?" we can develop a quantitative measure of stability. Essentially, a set of point pairs is stable if any small deviation from the correct alignment will cause the error to increase significantly. Assuming a set of points $\{\mathbf{p}_1, \ldots, \mathbf{p}_k\}$ with normals $\{\mathbf{n}_1, \ldots, \mathbf{n}_k\}$ corresponding to points $\{\mathbf{p}'_1, \ldots, \mathbf{p}'_k\}$ with normals $\{\mathbf{n}'_1, \ldots, \mathbf{n}'_k\}$ and examining the alignment computation under the point-to-plane error metric with linearized rotation, we see that the change in error ΔE due to changes in the rotational and translational alignment components $(\Delta\mathbf{r},\ \Delta\mathbf{t})$ is [Gelfand *et al.* 2003]:

$$\Delta E = \begin{pmatrix} \Delta\mathbf{r}^T & \Delta\mathbf{t}^T \end{pmatrix} C \begin{pmatrix} \Delta\mathbf{r} \\ \Delta\mathbf{t} \end{pmatrix}, \text{ where } C = \begin{pmatrix} \mathbf{p}_1 \times \mathbf{n}'_1 & \cdots & \mathbf{p}_k \times \mathbf{n}'_k \\ \mathbf{n}'_1 & \cdots & \mathbf{n}'_k \end{pmatrix} \begin{pmatrix} (\mathbf{p}_1 \times \mathbf{n}'_1)^T & \mathbf{n}'^T_1 \\ \vdots & \vdots \\ (\mathbf{p}_k \times \mathbf{n}'_k)^T & \mathbf{n}'^T_k \end{pmatrix}.$$

The six eigenvectors of the covariance matrix C give the natural decomposition of the alignment into orthogonal screw motions (translation along an axis coupled with rotation about it), and their eigenvalues $\lambda_1 \geq \cdots \geq \lambda_6$ give the stability of the alignment with respect to each eigenvector; a large eigenvalue indicates that change along the corresponding screw motion will induce a large increase in error. More importantly, if $c = \frac{\lambda_1}{\lambda_6}$ is close to one, *any* change in the alignment will yield a roughly comparable increase in error, indicating a well-formed minimum. Of course, the change in a point's error induced by rotation is proportional to the point's distance from the center of rotation. To balance the rotation and translation error, the set of point pairs is normalized to have its center of mass at the origin, with unit mean distance of a point from the origin.

For a pair of scans, we can compute C for the overlap region, using uniform random sampling. Next we can generate a new set of samples which constrains the different degrees of freedom more equally. To do this, we observe that the inherent stability of a point $\mathbf{p}$ with normal $\mathbf{n}$ with respect to itself is just [Brown and Rusinkiewicz 2007]:

$$P_{stability}(\mathbf{p}) = \begin{pmatrix} \mathbf{p} \times \mathbf{n} & \mathbf{p} \end{pmatrix} C^{-1} \begin{pmatrix} \mathbf{p} \times \mathbf{n} \\ \mathbf{p} \end{pmatrix}. \tag{2.1}$$

Here $\mathbf{n}$ is the normal of $\mathbf{p}$, not its correspondence, but since in a perfect alignment both $\mathbf{p}$ and $\mathbf{p}'$ will have the same normal, this still tells us how well any point pair containing $\mathbf{p}$ will constrain the alignment. Now we select points according to $P_{stability}$ in order to achieve a stable point set.

To illustrate intuitively how the stability distribution functions, consider once more the flat plane with a bump. For simplicity, we will assume this plane is the *xy*-plane (with the bump at the origin), and we will select a point $\mathbf{p}$ on it. Any rotation of $\mathbf{p}$ about the *x*- or *y*-axis will move it off the plane, as will a translation along the *z*-axis, increasing alignment error. $\mathbf{p}$ is therefore a good candidate to constrain these three degrees of freedom. On the other hand, any other point in the *xy*-plane will constrain them well also, so the eigenvectors of C corresponding to these motions will be large and $P_{stability}(\mathbf{p})$ will consequently be small. Now select a point $\mathbf{q}$ on the bump. This point will do a poor job of constraining rotations around the *x*- and *y*-axes (assuming it is spherical), but it will do an excellent job constraining translations along them. Since few points constrain these axes, the eigenvalues corresponding to them in C will be small, and $P_{stability}(\mathbf{q})$ will be large. By sampling according to $P_{stability}$ we will therefore be more likely to select points on the bump than on the plane surface, balancing out their relative paucity. In fact, we should end up with approximately half our samples on the bump, and half on the plane, constraining all degrees of freedom equally.

Gelfand *et al.* [2003] uses a slightly different version of stability sampling. Rather than evaluating the overall stability of each point, the authors measure the stability with respect to each degree of freedom by calculating $(\mathbf{p} \times \mathbf{n}, \mathbf{p}) \cdot \mathbf{x}_i$ where $\mathbf{x}_i$ is the *i*-th eigenvector of C. While this theoretically allows one to explicitly balance the different degrees of freedom, we have not found it to make a difference in practice. However, rather than sampling the stability distributions, they also make sorted lists of points with respect to stability on each degree of freedom, and take samples from the top of each list. For pathological cases, where the few truly stable points are overwhelmed by noisy measurements on the rest of the surface, sorting does help. The front faces of *Forma Urbis* fragments exemplify this situation: the shallow incisions of walls are the only features that constrain planar translation and rotation, but they are so sparse and shallow, that they will still be undersampled. We support this situation using a single list of points ordered by $P_{stability}$.

2.1.2 Distance Metrics

Let $\mathbf{p}$ be a point on scan A, and suppose the closest point to it on scan B is $\mathbf{p}'$. Then the point-to-point and point-to-plane distance metrics both approximate the squared distance function to B near $\mathbf{p}'$ with normal $\mathbf{n}'$. Pottmann *et al.* [2006] unifies these approximations in a second-order Taylor expansion of the squared distance function at $\mathbf{p}'$. Without loss of generality, we assume a basis centered at $\mathbf{p}'$ where the x- and y-axes correspond to the two principal curvatures directions of B at $\mathbf{p}'$, and the z-axis coincides with $\mathbf{n}'$. Then the squared distance function becomes

$$D(\mathbf{p}) = \left(\frac{d}{d-\rho_1}, \frac{d}{d-\rho_2}, 1\right) \cdot (p_x^2, p_y^2, p_z^2)$$

where $\rho_i = 1/\kappa_i$ is the inverse of the i'th principal curvature, and $d = |\mathbf{p} - \mathbf{p}'|$. Now as $\mathbf{p}$ gets progressively closer to $\mathbf{p}'$, $d \to 0$, $\frac{d}{d-\rho_i} \to 0$, and $D(\mathbf{p}) \to p_z^2$. However, since the z-axis coincides with n' in this basis, $p_z^2 = (\mathbf{p} \cdot \mathbf{n}')^2$, which is the squared distance under the point-to-plane metric. On the other hand, as $\mathbf{p}$ moves away from $\mathbf{p}'$, $d \to \infty$, $\frac{d}{d-\rho_i} \to 1$, and $D(\mathbf{p}) \to p_x^2 + p_y^2 + p_z^2$, which is the point-to-point metric since $\mathbf{p}'$ is at the origin in this basis. Therefore, when scans A and B are far apart, the point-to-point metric best approximates the squared distance function and provides the best convergence, whereas when they are in close alignment, point-to-plane is preferable. Going further, the authors describe how to efficiently compute the quadratic squared distance function using a data structure they call a d2tree. This improves the convergence properties of ICP when surfaces are not well aligned to start with.

2.2 Global Rigid Alignment

While ICP and its variants are widely used for pairwise alignments, they do not consider multiple scans simultaneously. Instead, an additional optimization is usually performed over all possible pairwise alignments. In this section, we review several approaches to global, or multiview, registration.

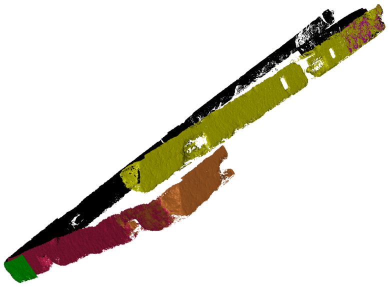


Figure 2.3: When scans are aligned sequentially, error accumulates, producing a gap between the yellow and orange scans rather than a closed loop.

This simplest form of global registration involves sequentially aligning each scan to the previous scan [Chen and Medioni 1992]. Aside from its simplicity, sequential alignment is fast: it requires only n ICP alignments to fully register a set of n scans. However, it suffers from accumulation of error. Given a sequences of scans $S_0, \ldots, S_n$, let ε_i be the alignment error between S_{i-1} and S_i. We define ε_0 to be the alignment error between S_n and S_0, which is not explicitly computed in sequential alignment. If alignment error is well distributed across the entire model, we will have $\varepsilon_0 \approx \varepsilon_1 \approx \cdots \approx \varepsilon_n$, but sequential alignment guarantees only that $\varepsilon_0 \leq \sum \varepsilon_i$. The accumulation of error in ε_0 is most apparent when a sequence of scans should form a closed loop; with sequential alignment a large gap often remains between the first and last scans.

Overcoming the error accumulation problem requires a global optimization in which all overlapping scans are aligned to each other. One way to do this is to pick a scan, and compute its least-squares alignment to *all* scans simultaneously. This operation is performed iteratively for all scans until the alignments converge. Since each step must decrease the alignment error associated with a particular scan, convergence is guaranteed, albeit slow. The process can be accelerated by prerecording point correspondences between all pairs of scans, and reusing them at each iteration. Assuming the initial alignment is good enough that pairwise ICP can correctly align all pairs of scans, this approach has two advantages: it improves performance by eliminating closest point computations from the inner loop, and it reduces the memory footprint to a global set of point correspondences. For very large models, such as the Digital Michelangelo models, it is impossible to keep all scan data in memory, making the latter property especially important. Several variations of this algorithm exist: it is possible to compute new alignments for each scan based on the current alignments, then update all transformations at once [Bergevin *et al.* 1996]; alternatively, scan alignments can be updated as they are computed [Benjemaa and Schmitt 1998; Williams and Bennamoun 2000]; finally, scans can be added to an active set one at a time, with global optimization performed over all active scans before a new scan is added [Pulli 1997].

Rather than optimizing the alignment one scan at a time, as in the previous methods, it is possible to optimize multiple transformations at once, which may help avoid local minima in the error function. Sharp *et al.* [Sharp *et al.* 2004] optimizes *cycles* of scans until convergence. To do so, the authors treat each scan as a node in a graph, with scan overlap corresponding to a graph edge. They determine all graph cycles which can be created by adding a single edge to the graph's minimum spanning tree, and show how to optimize the alignment of all scans in the cycle. At each iteration, new alignments are computed along each cycle. For edges that belong to multiple cycles, the different alignments are averaged. The authors show that this procedure must reduce total error at each iteration, and therefore converge. Intuitively, this algorithm directly addresses the problem of accumulated error around closed loops.

Other methods directly optimize the total alignment error. The global alignment error function is of course more complex than the error associated with a single scan, but tackling all alignments at once can help avoid local minima by directly updating all alignments at the same time. Neugebaur [1997] solves for the translational component in closed-form, then uses a hierarchal Levenburg-Marquardt approach to optimize over rotation angles. Krishnan *et al.* [2005] shows how to optimize directly over rotation matrices rather than angles. The challenge is that the tangent planes to the error function do not themselves lie in the space of rotations, so it is necessary to reproject the new alignments onto the rotation space at each iteration. Pottmann *et al.* [2006] performs the same optimization using their point-to-surface distance metric.

For a dataset of n scans, each of which overlaps D other scans on average, the number of variables involved in the global optimization of rotation is $O(nD)$, and solving for the translation requires a singular value decomposition on an $n \times n$ matrix. As n increases, the number of variables becomes intractably large. Local optimizations, although they take many more iterations to converge, are therefore preferred in these cases (such as with Digital Michelangelo models).

A slightly different optimization arises if a consistent set of correspondences are used across all scans [Wen *et al.* 2005]. Each scan i contains a set of points $\{p_k^i\}$, with all p_k's corresponding to each other. In this case, a single target location q_k is chosen for each p_k (initially as the locations on the first scan, for example), and each scan is aligned to the target locations. Each q_k is updated to the average of all p_k's, and the process iterates until convergence. While it is unusual to use a consistent set of point correspondences in the range scanning context, the non-rigid alignment technique we present in Chapter 3 does so in order to more easily enforce global constraints on deformation. Although we take a different approach to determining target locations, we retain the idea of independently aligning each scans to those positions.

A final approach to global, rigid alignments is alternately reconstructing a complete model, to which individual range scans are realigned [Huang *et al.* 2007]. Only $O(n)$ alignments per iteration are required in this case. More importantly, each scan is simultaneously aligned to information from all scans, which improves stability in situations where one scan has many small overlap regions with different scans. Further, when aligning to a union of range scans, the source scan will be biased toward the currently closest target scan; by aligning to a merged surface which has already averaged out measurement noise and alignment error, bias is avoided. However, surface reconstruction is computationally expensive, and therefore impractical in any context where performance is a consideration.

2.3 Pairwise Non-Rigid Alignment

Non-rigid alignment techniques arise in both the range scanning and medical registration contexts. In both cases, they are typically geared to the registration of moving objects, or of a scanned object to a template. Usually only two scans or objects are involved, and global alignment is unnecessary; the deformation is assumed to be large, and high-frequency details are not expected to align. Existing algorithms have therefore made design decisions which emphasize great flexibility in the allowable deformation. In contrast, the deformations that occur in large-scale range scanning applications are small and low-frequency, it is essential that high frequency detail align precisely, and a scalable, global alignment is imperative.

Hähnel *et al.* [2003] and Allen *et al.* [2003] consider the problems of registering scans from moving objects and of registering full body scans to a template model. Both replace the rigid-body transformation of ICP with a per-vertex affine transformation and smoothness constraint to control the variation in transformation between neighboring vertices. As a result, they must ultimately consider *every* vertex of the source scan (although multi-resolution optimizations mitigate the impact at early iterations), and the rigid-body computation is replaced by a high-dimensional non-linear optimization. The former work also covers global alignment, however a framework similar to Neugebauer [1997] is necessary to ensure both deformation and error are evenly distributed, further limiting the manageable problem size.

The medical imaging field has also long used non-rigid registration techniques to align medical scans (often X-rays or MRIs). Often, the objective is to align a scan to an organ template, to aid in identifying and labeling structures, to align a recent scan to an older one of the same subject, or to align different types of data to each other. As in the preceding cases, these are pairwise alignment problems, deformations will be large, and high frequency detail will be different between scans. Rather than apply a dense transformation, a sparse set of markers is extracted, and a low frequency transformation is computed.

One of the more popular functions for medical registration is the thin-plate spline, described in Section 3.1.5, which we have also found practical for range scans. It computes a deformation that maps a point set X to a corresponding point set Y, while minimizing the total deformation. In particular, the thin-plate spline reduces to an affine transformation whenever possible. An additional smoothness parameter λ trades off the spline's precision at mapping X to Y with the amount of warp. It is not possible to directly substitute a thin-plate spline for the rigid-body transform in ICP because any set of points correspondences can be fit exactly — ICP would achieve zero error at the first iteration, but with high warp! However, a similar iteration maps each source point to a Gaussian-weighted average of target points, using a high smoothness term (so the targets are only roughly approximated). At each iteration, the width of the Gaussian for averaging is narrowed, and the smoothness term is reduced, until convergence [Chui and Rangarajan 2003]. While practical for relatively small, sparse point sets, the cost of computing many thin-plate splines quickly becomes prohibitive, and we have found that the computation becomes numerically unstable when the sample-spacing within a scan is on the order of the desired alignment precision. Therefore, for dense range scans, alternative methods are necessary.

We have previously developed an approached to pairwise, non-rigid alignment of range scans which also represents the warp using thin-plate splines [Brown and Rusinkiewicz 2004]. We first dice the source scan, and rigidly align each piece to the target scans, in order to compute accurate correspondences. Since we assume a low-frequency warp, small pieces will align well with a rigid-body transform. Then, using these correspondences, we compute a thin-plate spline which aligns the source scan to the target. However, there is a tradeoff between the size of each piece and its alignment stability. As a result, point correspondences near the center of a piece are more accurate than those near the edge, and the dicing structure is evident in the computed warp. In addition, there is no mechanism to distribute the warp evenly among many scans, which is necessary to support global alignment.

2.4 Global Non-Rigid Alignment

Although most work on non-rigid alignment has been restricted to the pairwise case, their have been efforts at global alignment aimed specifically at range scans. Because of its additional degrees of freedom, global non-rigid alignment can achieve zero error at the expense of high warp. Clearly, this would not meet our goal in range scan alignment; any warp we apply that does not compensate for a warp in the data will degrade our final model. Therefore, a global, non-rigid alignment algorithm must minimize and evenly distribute both error and warp. In the figure at the right, the peach scan has been simultaneously aligned to the blue and red scans using non-rigid alignment pulling in two directions at once, which leads to stretching and high-frequency artifacts such as the pinched region on the nose.

In the Digital Pietà Project [Bernardini *et al.* 2002], conformance smoothing was developed to handle error in depth estimates. Since the position estimates were reliable (that is, the line of sight from the camera center through each pixel was correctly estimated), conformance smoothing adjusts each vertex *along* the line of sight to improve rigid-body alignment accuracy to other range scans. However, calibration error affects the position estimates, so any compensation must adjust the lines of sight as well.

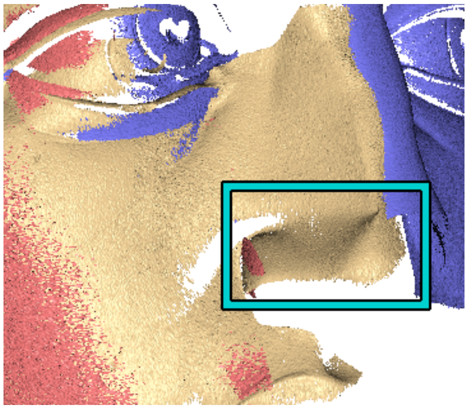


Figure 2.4: The peach scan has been simultaneously aligned to the blue and red scans using non-rigid alignment, leading to stretching and high-frequency artifacts such as the pinched region on the nose.

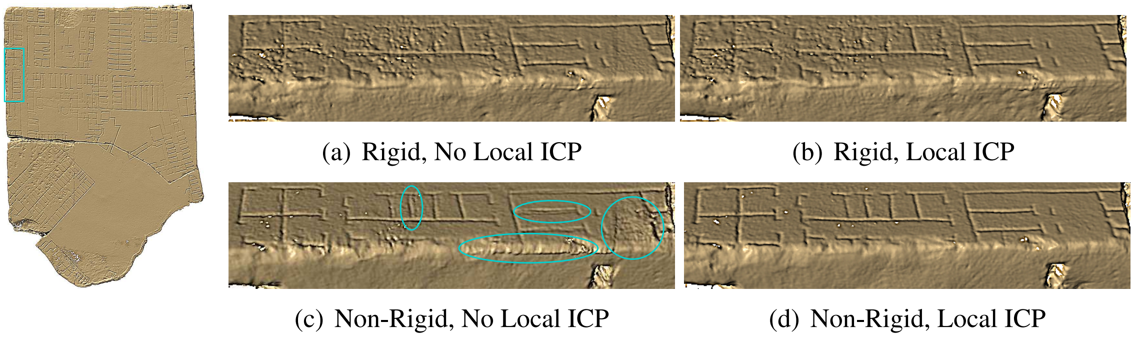
Hierarchical ICP [Ikemoto *et al.* 2003] addresses the calibration problem by dicing each range scan into small, overlapping pieces, and rigidly aligning all the pieces. This works because the deformation is low frequency, and therefore is well approximated locally by a rigid-body transform. Aligning overlapping pieces of the same scan to each other as well as to all other scans constrains the transformations of neighboring pieces to be similar. However, the degree of overlap and size of pieces must be carefully chosen: if the overlap region does not contain enough detail to constrain alignment, the two pieces can move independently of each other, creating gaps; if there is too much detail in the overlap region, they will lock strongly onto each other and effectively operate as a single fragment. The overall size of each fragment has a similar effect: too small and there will not be enough detail for a stable alignment; too large and there will be too much deformation to approximate with a rigid-body transformation. Even when the size and overlap of fragments is well chosen, the fact the pieces of the same scan are allowed to misalign with themselves guarantees some smoothing of high-frequency details will occur in the final model. Additionally, this method is prohibitively slow — recall that global alignment is $O(nD)$ in the number of scans and degree of overlap; dicing scans increases both.

Dynamic geometry registration [Mitra *et al.* 2007] also uses piecewise rigid transforms, but uses a regularization framework to construct a smooth deformation field. However, other design choices reflect its goal of aligning scans from high frame-rate scanners. In particular, it aligns scans using a space-time gradient similar to optical flow methods, which requires that successive scans differ only slightly. It is also fundamentally sequential, rather than fully global, in nature, so it will not distribute warp evenly across all scans. These design choices are appropriate for real-time alignment using high frame-rate scanners, but not for the large objects in which we are interested.

Chapter 3

Global Non-Rigid Alignment

In this chapter we investigate the problem of scanning very large objects. As we discussed earlier, triangulation scanners have error proportional to viewing volume (or equivalently to the distance of object points from the scanner). Because scans must be merged and retriangulated to obtain a final mesh, even small misalignment leads to smoothing of high-frequency details in the merged mesh. Larger misalignments may result in corrupt reconstructions. In the case of data acquired by the Digital Michelangelo Project [Levoy *et al.* 2000], the desired precision of approximately 0.25 mm combined with viewing


(a) Rigid, No Local ICP

(b) Rigid, Local ICP

(c) Non-Rigid, No Local ICP

(d) Non-Rigid, Local ICP

Figure 3.1: Using non-rigid global alignment techniques, we are able to preserve more surface detail with fewer merging artifacts, as seen in closeups of the upper-left edge of this *Forma Urbis Romae* fragment. *(c)* Using non-rigid alignment (section 3.1.3) improves alignment enough to eliminate most surface noise. *(d)* Incorporating locally weighted ICP-based feature correspondences (section 3.1.2) further improves detail alignment, notably eliminating double edges.

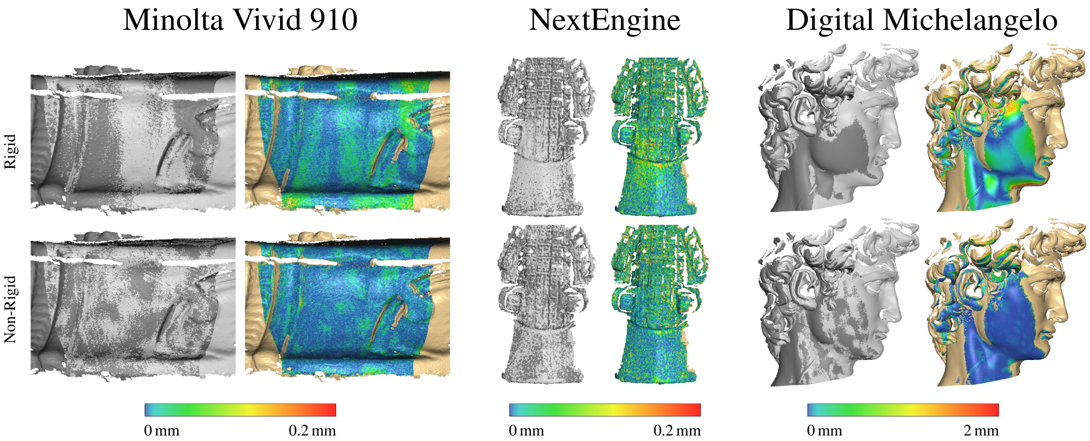


Figure 3.2: Rigid vs. non-rigid alignments of range scans from different 3-D scanners. The gray figures show two range scans (light and dark) superimposed, while the color figures show the alignment error between the two scans. Low-frequency warp in the initial range scans leads to biased alignment error, which in turn causes one scan to be in front of the other over large contiguous regions. Since the non-rigid alignments compensate for this warp, bias is removed, and neither scan appears consistently in front of the other. Non-rigid registration also reduces alignment error across the scan, as shown in the color figures.

volumes over 5 m made perfect scanner calibration unattainable, leading to warping on the order of several millimeters. Even much smaller scanners, which acquire similar-resolution geometry in a much smaller working volume, exhibit warping when their resolution is pushed to the maximum (Figure 3.2). Over time, we believe that calibration will get harder, not easier, because sensor resolution is increasing faster than lens and motor precision.

We propose a method for simultaneously computing non-rigid warps to a large collection of 3D meshes, in order to minimize scan-to-scan distances in *all* overlapping regions. The alignment minimizes the warp applied to each scan, and distributes error evenly across all meshes. By correctly aligning surface details, the method ensures that low-frequency errors do not corrupt high frequencies of the reconstruction: the merged models exhibit less smoothing and fewer artifacts than their rigidly-aligned counterparts (Figure 3.1).

Our registration goals are high alignment accuracy and efficiency on large datasets. Specifically, we preserve the overall shape of the scans, which is generally correct, while

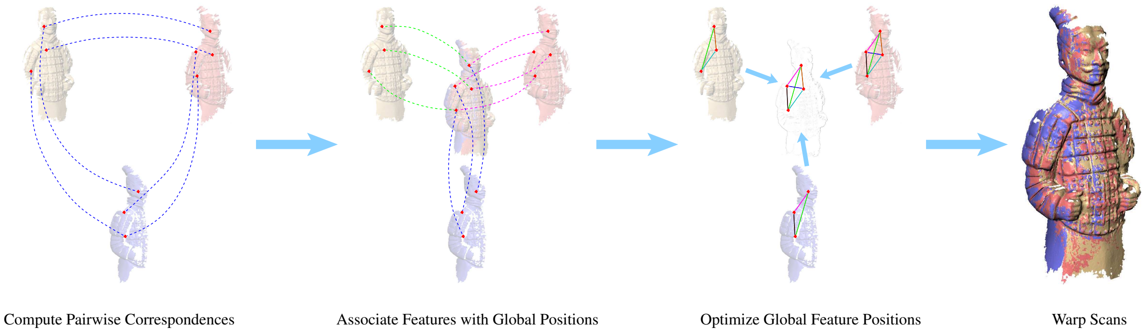

Compute Pairwise Correspondences
Associate Features with Global Positions
Optimize Global Feature Positions
Warp Scans

Figure 3.3: The global alignment framework. We compute pairwise correspondences between features selected on each range scan and all other range scans. From these we obtain a set of global features whose positions on each range scan are known. Next we optimize the locations of these features using relative distances between them as constraints. Finally, we warp and align each range scan using the features as control points of a thin-plate spline.

improving the alignment of surface detail so that the final model is sharper. Since the warp is low frequency, we use a relatively small number of control points, which improves performance and helps prevent high-frequency distortions. We also avoid performing matrix operations such as singular value decomposition on the pairwise constraint matrix so that we can scale to arbitrarily large data sets. Our main contributions are the following:

1. *Feature correspondences*: We introduce a novel variant of iterative closest points (ICP), locally weighted ICP, for efficiently finding correspondences between deformed ranged scans.

2. *Global optimization*: Our global optimization framework is highly parallelizable and memory efficient, making it practical for arbitrarily large datasets. It is agnostic to the final transformation, so that it is suitable for rigid as well as non-rigid alignment.

3. *Thin-plate spline warp*: We show how to use non-rigid thin-plate spline warps to consistently warp many scans simultaneously. Their use in range scan alignment has previously been limited to pairs of range scans.

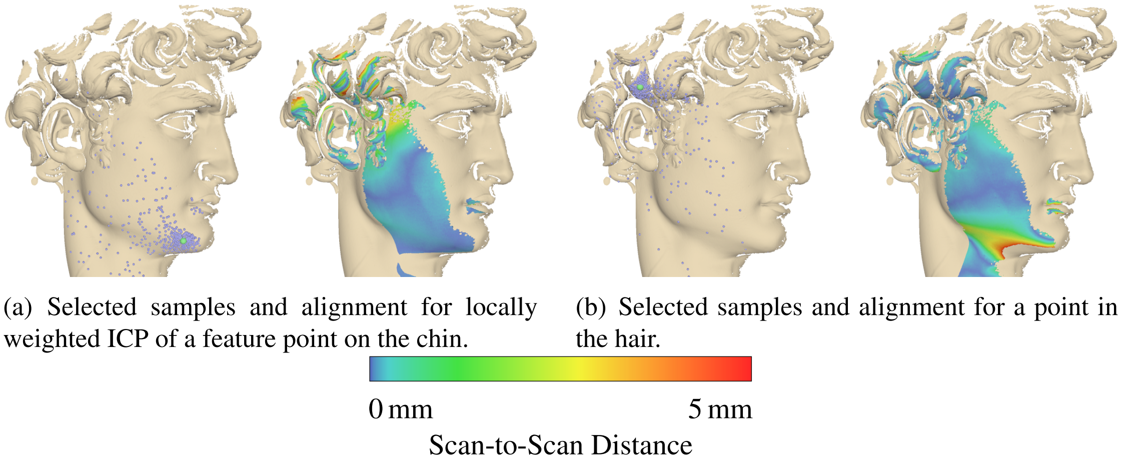
3.1 Global Alignment Framework

Global alignment aims to find alignments for all scans which minimize overall alignment error while distributing it evenly among scans. Significant error generally results in smoothing and/or artifacts in the final mesh; since low-frequency error in individual range scans produces substantial rigid-body misalignment, it effectively causes error at *all* frequencies in the merged result. In the non-rigid case, the undesirable high frequency misalignment can be traded for a low frequency warp in the range data, restricting the final error to the frequencies where error already existed. On the other hand, as discussed in Section 2.4 it is possible to produce a badly warped global alignment which nevertheless has low alignment error, so it is important to minimize and evenly distribute the warp.

We solve the problem of minimizing and distributing warp (Figure 3.3) by first finding the locations of a common set of features on all range scans, using the locally weighted ICP method described in Section 3.1.2. Conceptually, we might find the locations of the warrior's mouth, elbow, shoulder and collar bone in every range scan. We next find positions in space for each feature points such that all range scans can be mapped to these locations with minimal warp. Finally, using the features as control points, we compute a thin-plate spline warp for each range scan that brings it into alignment with the global point positions (and hence with all other range scans). Note that, as in most previous work on global registration, we assume that an approximate initial alignment is available.

3.1.1 Feature Point Selection

Feature points must cover the entire model and yield stable correspondence computations between range scans. While it may be intuitive to think of these features as derived from the object (*e.g.* the tip of David's nose), we do not select them this way because we have


(a) Selected samples and alignment for locally weighted ICP of a feature point on the chin.

(b) Selected samples and alignment for a point in the hair.

Figure 3.4: Locally weighted ICP focuses the alignment on a small region, producing a better result in that area and a worse one elsewhere. We use locally weighted ICP centered at feature points to compute accurate correspondences.

only the initial range scans available at this stage. Instead, we select features on each range scan, then find their correspondences on every other range scan. We select half the features using uniform random sampling (to ensure coverage in smooth areas of the model), and half using the stable sampling technique described in Section 2.1.1 (to ensure we can find good feature correspondences wherever possible).

Since the feature points will eventually define a spline, the number of points required depends in principle on the degree of warp in the scans. In practice, however, we have found the algorithm to be relatively insensitive to the number of points that are selected. Instead, we sample a fixed percentage of points from each range scan, typically 1% for small models, 0.1% for medium-size models, and less for very large models with a high degree of overlap, such as the Digital Michelangelo scans.

3.1.2 Locally Weighted ICP for Correspondences

The next stage of our pipeline is to find the correspondence for each feature point $\mathbf{f}_i$ on each scan that overlaps it. In order to find correspondences efficiently for thousands of points distributed across many scans, we use a novel variant of the ICP algorithm. Although ICP is usually used for aligning entire scans, it has also been used on "diced" sections of scans to provide more local alignment [Ikemoto *et al.* 2003; Brown and Rusinkiewicz 2004]. We extend this idea with "locally weighted ICP," which gives higher weight in the registration to the portion of the mesh immediately surrounding a feature $\mathbf{f}_i$. After alignment, the closest point on the other mesh is used as the correspondence.

Our ICP variant is based on the "baseline" algorithm described by Rusinkiewicz *et al.* [2001], with a point-to-plane error metric, rejection of matches to edge points, closest-point computations accelerated with a *k*-D tree, and matching based on compatibility of normals. For additional stability, we also use compatibility of mean curvatures to improve matching.

Our major change is to the point-selection stage: source points are sampled using a probability distribution that makes it more likely that points near $\mathbf{f}_i$ are used. The major component of the probability is a decreasing function of distance from the feature point $\mathbf{f}_i$ under consideration:

$$P_{feature}(\mathbf{x}) = \frac{1}{\varepsilon + \|\mathbf{x} - \mathbf{f}_i\|^2}. \tag{3.1}$$

If, however, the weight were based entirely on this, there would be a danger of selecting only points that lie on a region of the mesh without sufficient geometric variation to constrain all six degrees of freedom of a rigid-body transformation. Therefore, we augment the probability function with the $P_{stability}$ term from Section 2.1.1 that assigns high probability to locations on the mesh that constraint the alignment. (For the *Forma Urbis Romae* dataset we sort the probability distribution and use the best points, again for the reasons discussed in Section 2.1.1.

Our final probability function consists of the product of $P_{feature}$ and $P_{stability}$. The function is normalized, integrated into a cumulative distribution function, then numerically inverted to transform a uniform random variable into samples to be selected (*i.e.*, the “inversion method” for importance sampling). A separate locally weighted ICP is run for each feature point $\mathbf{f}_i$, and the nearest point on the target range scan is selected as the correspondence to $\mathbf{f}_i$. Figure 3.4 shows the results of aligning a pair of scans for two different feature points. Note that the points used for ICP are selected in the vicinity of the feature points, leading to lower alignment errors in those regions, and therefore more accurate correspondences.

Our feature correspondence algorithm adds little to the total computation time, relative to the time that would be required for rigid-body registration. Because an initial overall alignment of each pair of range scans is performed first, it takes only a few iterations of ICP (taking a few milliseconds) to perform each weighted alignment. The most expensive parts of the ICP, building *k*-D trees and computing the overlapping region between each pair of meshes, are computed only once per pair. Per-vertex normals and curvatures are generated and smoothed (to remove noise) as a preprocess, so this computation is performed only once per range scan.

3.1.3 Global Point Positioning

For a given feature i, we denote the location of its correspondence on mesh m by $\mathbf{f}_i^m$. In order to compute a consistent alignment, we need to find a “global position” $\mathbf{g}_i$ for each feature point, based on the known correspondences $\mathbf{f}_i^m$. Moreover, we must find global positions that do not induce unnecessary warp, as in Figures 2.4 and 3.5. Intuitively, for any mesh m, the mutual arrangement of all feature locations $\mathbf{f}_i^m$ should differ as little as possible from the spatial arrangement of the corresponding $\mathbf{g}_i$. If all correspondences are correct, and there exists a rigid-body alignment of all scans, then these spatial arrangements should be identical; in the presence of a low frequency warp this will be almost true, with nearby features maintaining their spatial arrangement better than distant ones.

Two simple (albeit inadequate) heuristics for computing the position of each $\mathbf{g}_i$, illustrate the relationship between each $\mathbf{g}_i$ and the corresponding $\mathbf{f}_i^m$. The first places each $\mathbf{g}_i$ at the location where the feature was originally selected. By this, we mean that if feature i was initially chosen among the vertices of mesh m, then $\mathbf{g}_i = \mathbf{f}_i^m$. Unless the range scans were perfectly aligned to start with, this will result in mutually inconsistent positions for the $\mathbf{g}_i$, and we will have a situation akin to Figure 2.4. Alternatively, we could set the position of $\mathbf{g}_i$ to be the average over all $\mathbf{f}_i^m$. This too will fail, since even nearby features have correspondences on different sets of range scans (Figure 3.5), again resulting in mutually global positions.

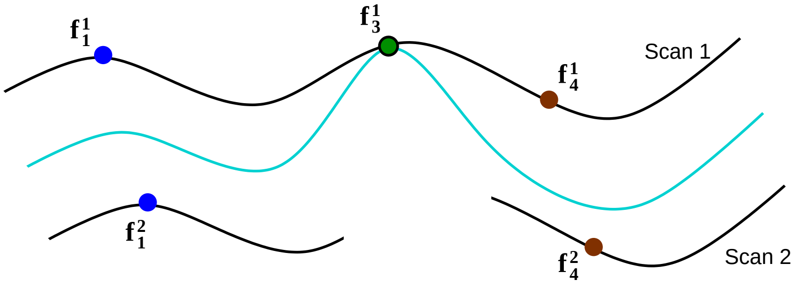


Figure 3.5: Mutually consistent global feature positions must be selected to prevent unnecessary warp from being introduced. Here the average position for each feature was used; these are not mutually consistent because the green feature has no correspondence on the bottom scan.

Instead, guided by the intuition that global point positions should be affected by their geometric configuration relative to neighboring features, we optimize for the $\mathbf{g}_i$ by attempting to preserve their relative distances, as computed within each mesh. This is in some ways similar to the approach of Laplacian mesh coordinates [Sorkine *et al.* 2004], in which each vertex is represented relative to its neighbors. Conceptually, we consider placing a "spring" with *non-zero rest length* between all pairs of features i and j. Assuming feature i was originally selected on mesh m and feature j on mesh n, we set this rest length to be $\frac{1}{2}(|\mathbf{f}_i^m - \mathbf{f}_j^m| + |\mathbf{f}_i^n - \mathbf{f}_j^n|)$ (Figure 3.6).

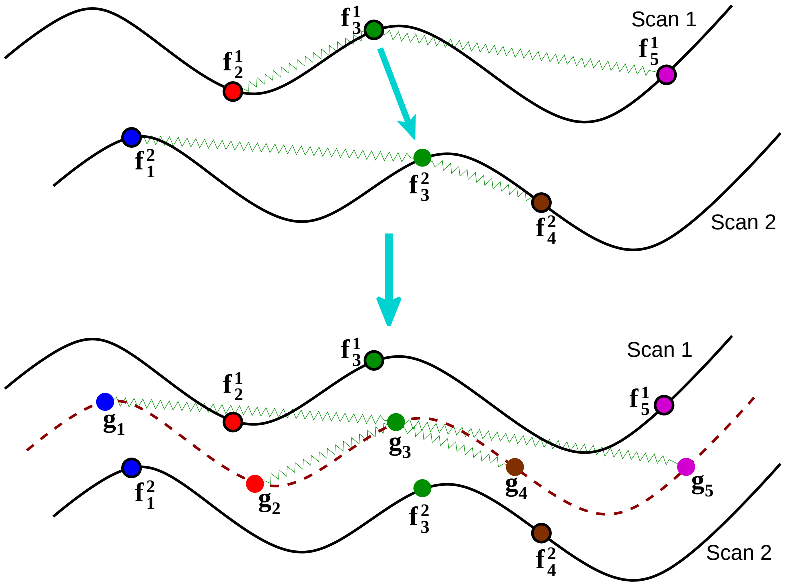


Figure 3.6: *Above:* Global feature points are attached to each other by springs whose desired length is based on distances along range scans. Since the red ($\mathbf{f}_2$), green ($\mathbf{f}_3$) and magenta ($\mathbf{f}_5$) features are all associated with the top scan, the rest length of the springs attaching them is the actual distance between the features. However the rest length of the springs attaching $\mathbf{f}_3$ to the blue ($\mathbf{f}_1$) and brown ($\mathbf{f}_4$) ones is based on the distance from the green feature's *correspondence* on the bottom scan. *Below:* Point positions are adjusted to minimize the total spring energy, ensuring that the aligned meshes will have minimal warp.

Mathematically, we form an error metric for each point pair:

$$\mathcal{E}_{ij} = \sum_m w_{ij} \left(|\mathbf{g}_i - \mathbf{g}_j| - |\mathbf{f}_i^m - \mathbf{f}_j^m| \right)^2, \tag{3.2}$$

where the weighted sum is taken over all meshes m such that either:

- Both features i and j were originally selected on mesh m, or
- Feature i was selected on m and feature j has a valid correspondence $\mathbf{f}_j^m$ on m.

The overall energy function is the sum of all the pairwise functions. To minimize it, we initialize each $\mathbf{g}_i$ to be the average of all valid $\mathbf{f}_i^m$, then move one point at a time using gradient descent. Despite the simplicity of this approach, we have found it to be efficient and robust (in particular, it is more stable than Newton's method), though of course it is not guaranteed to converge to a global minimum.

3.1.4 Outlier Rejection

Our global point positioning approach shares the drawback of all least-squares methods: it is sensitive to outliers. Incorrect correspondences, which occur most commonly in flat areas, lead to incorrect springs and poor feature positioning. Therefore, we implement aggressive pruning of outlier correspondences, based on the error and stability information we obtain from ICP, both in isolation and in relation to nearby features.

The first stage of outlier rejection is based on absolute error thresholds. If the unweighted ICP we perform between a pair of scans before computing correspondences fails (due to lack of stability or insufficient overlap), we do not record any correspondences between that pair of meshes. We also reject any correspondence (1) whose RMS ICP error (point-to-plane distance) is above a threshold (30 mm), (2) which is more than 50 mm from the average feature location, or (3) whose stability ($P_{stability}$ as reported by locally-weighted ICP) is lower than 0.001. These thresholds are only designed to eliminate egregious correspondences, and are therefore set to be large relative to the mesh sample spacing.

After optimizing feature point locations, we do additional pruning of points based on aggregate error. The first step is to thin out feature points, so that no two features are closer than some threshold to each other. Starting from each remaining feature point, we find all other feature points within this distance, and discard all but the one with lowest spring energy. This helps prevent high-frequency artifacts due to positioning errors, but more importantly it reduces the running time of computing thin-plate splines. The appropriate threshold to use depends on the sampling density and size of the dataset. For Digital Michelangelo data we use a minimum distance of 8 mm, while for small datasets such as the penguin we allow points to be closer — about 2.5 mm.

Next we discard features which move too far relative to their nearby features on the same mesh. For each mesh m, we consider the list of features with correspondences on that mesh. For each of these features $\mathbf{f}_i^m$, we find the 8 nearest features (again with correspondences on the mesh), and calculate the median distance d between the feature locations on mesh m and their global positions. If $|\mathbf{g}_i - \mathbf{f}_i^m| > 4d$, we discard feature i. After all of these pruning steps we re-optimize the locations of the remaining feature points.

3.1.5 Warping with Thin-Plate Splines

Once the global features are positioned, all range scans must be warped to align to them. That is, we compute the warp that maps all the $\mathbf{f}_i^m$ on scan m to their global positions $\mathbf{g}_i$, and apply that warp to the entire scan. (Since these warps are computed independently for each scan, we drop m from our notation in this section, referring instead to $\mathbf{f}_i$.) The nature of the warp can be complex, difficult to model, and vary between data sets. For instance, calibration error, lens distortion, combining input from different sensors, and object movement all cause different kinds of warp. Each of these is difficult to model on its own, and we would like to handle all of them seamlessly. As in a number of other non-rigid alignment systems, such as Brown and Rusinkiewicz [2004] and Chui and Rangarajan [2003], we turn to the thin-plate spline. However, we extend the use of the thin-plate spline to global registration by mapping each feature point onto its *global* position.

Here we briefly summarize the properties of the thin-plate spline, originally introduced by Duchon [1977]. It is a non-rigid, globally smooth function that is easily computable, separable into affine and non-affine components, and contains the least possible non-affine warping component to achieve the mapping. By the last statement, we mean that the sum of squares of all second order partial derivatives is minimized. So, if $\mathcal{S} : \mathbf{R}^n \to \mathbf{R}$ is an n-dimensional thin-plate spline, the *bending energy*,

$$J = \int \left(\sum_{i,j} \mathcal{S}^2_{\mathbf{f}_i \mathbf{f}_j} \right) \mathrm{d}\mathbf{f}_1 \ldots \mathrm{d}\mathbf{f}_n \tag{3.3}$$

is minimal. Note that since affine transformations are linear, they contribute no error under this metric.

Duchon proves that, for two corresponding point sets $F = (\mathbf{f}_1 \ldots \mathbf{f}_n)^t$ and $G = (\mathbf{g}_1 \ldots \mathbf{g}_n)^t$, there is a unique function $\mathcal{S}$ such that $\mathcal{S}(\mathbf{f}_i) = \mathbf{g}_i$ and whose bending energy is minimal. Furthermore, this function takes the form $A\mathbf{x} + WK(\mathbf{x})$, where $\mathbf{x}$ is a point written in homogeneous coordinates, A is an affine transformation, W is a fixed n-dimensional row vector of non-affine warping parameters constrained to $WF^t = 0$, and $K(\mathbf{x})$ is an n-dimensional column vector where $K(\mathbf{x})_i$ is the Green's function $U(|\mathbf{x} - \mathbf{f}_i|)$. In our case (minimizing second order partials in $\mathbf{R}^3$), U is simply $|\mathbf{x} - \mathbf{f}_i|$; the constant factor is implicitly folded into W.

Thin-plate splines need not be interpolating. Instead, they can minimize the energy functional [Duchon 1977; Wahba 1990]

$$E_\lambda = \frac{1}{n} \sum |\mathbf{g}_i - \mathcal{S}(\mathbf{f}_i)|^2 + \lambda J \tag{3.4}$$

The spline will not be interpolating in this case, but for any fixed λ, there will still be a unique minimum, of the form described above.

For the interpolating case, the thin-plate spline specification provides a linear system of equations. For the approximating case, [Wahba 1990, eqs. 2.4.23 and 2.4.24] derives a similar system of equations by rewriting Equation 3.4 in matrix form, performing a QR decomposition on $\mathcal{S}$, and simplifying:

$$\begin{aligned} AF + W(K + n\lambda I) &= G \\ WF^t &= 0 \end{aligned} \tag{3.5}$$

where K is $n \times n$ and $K_{ij} = U(|\mathbf{f}_i - \mathbf{f}_j|)$.

Because we would like very precise alignments (since multiple measurements of rigid objects should align precisely), the spline must be heavily weighted toward interpolation. For this reason we must rely on accurate correspondences to produce a good alignment. Nevertheless, we set λ to a very small value (10^{-10}), because doing so improves the numerical stability of Equation 3.5 when many point pairs are used.

3.2 Results

We present alignment results from a variety of sources. The Digital Michelangelo scanner acquires range scans up to 50 cm long, with a warp of several millimeters. The commercially available NextEngine and Minolta Vivid scanners have much lower warp for a similar resolution, but acquire commensurately smaller viewing volumes. We also show applications of our methods to a photometric stereo dataset, which exhibits a different and much larger warp.

3.2.1 Digital Michelangelo Scanner

Michelangelo's David: Figure 3.8 shows closeups of the head of Michelangelo's David, created with global rigid-body alignment, and with our non-rigid alignment. Figure 3.7 shows cross sections of the range scan positions for each range scan at the right edge of the eye after ICP alignment and after our non-rigid alignment, as well as histograms of

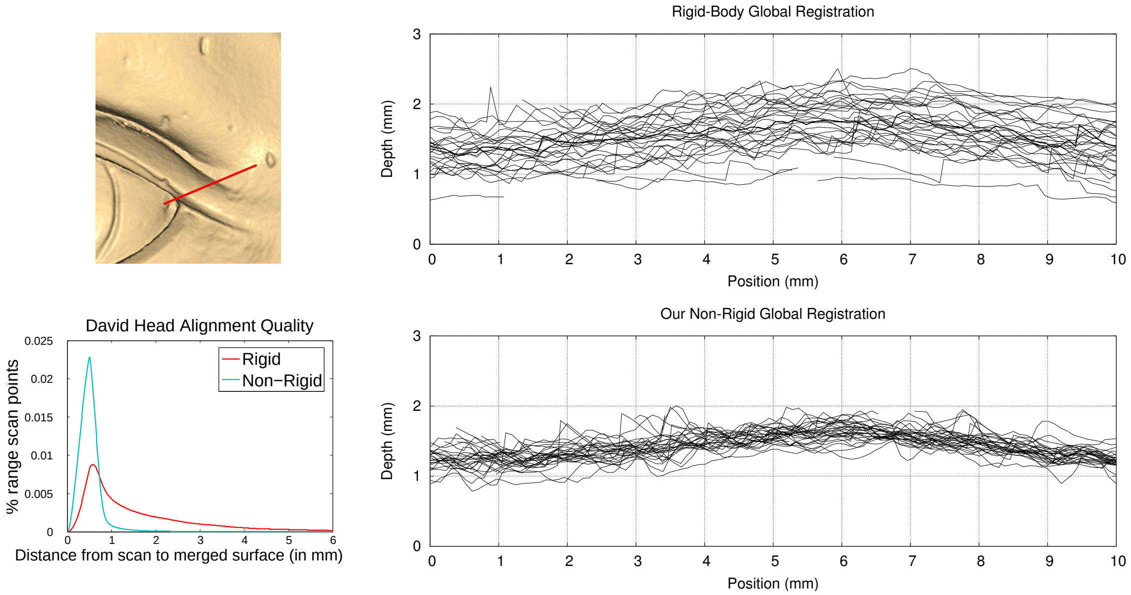


Figure 3.7: Alignment error plots for David's head. The line plots at right show the depths of the aligned range scans along the marked line on David's eye. The larger spread on the rigid scans *(above)* compared to the non-rigid ones *(below)* causes the smoothing and artifacts seen in figure. At the lower left are histograms of alignment error under rigid-body and non-rigid registration. The rigid-body alignment histogram reveals a long tail of misalignments by several millimeters, which corresponds to the greater spread of range scans seen in the depth plots.

alignment error under rigid and non-rigid alignments. The original data set for David's head is 1406 range scans, and the final model contains 28 million vertices when merged at 0.25 mm precision with VRIP [Curless and Levoy 1996].

The most obvious difference between the rigid and non-rigid models is the reduction in merging artifacts and smoothing due to misalignment. This is especially apparent in areas of high detail, such as the closeup of David's hair shown in Figures 3.8(e) and 3.8(f). Here, some of the chisel marks between locks of hair are badly smoothed in the rigid-body version, while others are nearly obliterated. The non-rigid version is sharper and clearer. More subtly (and counter-intuitively), rigid-body alignment can distort shapes, as it has in the pupil of David's right eye (Figure 3.8(a)). This occurs because the merging process follows different misaligned range scans in different areas. With nonrigid alignment this artifact is removed, and the local shape is better preserved.

We sampled 0.0075% of the range scan vertices to use as features, resulting in 16499 feature points, of which 5832 were retained for global registration. Much of the sample rejection comes during the point thinning stage, so that increasing the sampling rate would not improve alignment quality. Correspondence required 78 hours of CPU time, distributed across 60 nodes of a cluster (1.5 hours of wall clock time), while global point positioning required an additional 30 minutes on a single processor.

Forma Urbis Romae: Figure 3.1 shows piece 033 from the *Forma Urbis Romae* data set [Levoy *et al.* 2000]. Because the only features available on the top of these scans are $^1/_2$ mm-deep incisions, alignment is particularly tricky. Furthermore, scans taken at an angle so as to capture both the top and sides of the pieces cannot generally capture the incision due to self-occlusion; instead these scans contain holes with noisy boundaries. Because no full model of this piece previously existed, we have generated four different versions using our algorithms, ranging from a classic rigid-body alignment to the full non-rigid treatment. For Figures 3.1(a) and 3.1(c) we disabled locally weighted ICP; a single unweighted ICP alignment is computed for each pair of range scans, but no additional locally weighted ICPs are performed before reading off correspondences. This is similar to the pairwise alignments/correspondences computed in other multiview rigid-body algorithms. Similarly, for Figures 3.1(a) and 3.1(b), we did not compute thin-plate spline transforms. Instead, we perform our usual global point positioning, but compute only the best rigid-body alignment of each scan.

As is evident in Figures 3.1(a) and 3.1(b), rigid-body alignment leads to substantial misalignment of the top, resulting in surface roughness. Harder to see are the phantom incisions due to misalignment, which show up in both rigid-body alignments and when locally weighted ICP is removed. Some of these are circled in 3.1(c). We have found that the top is consistently better with weighted ICP correspondences and thin-plate spline alignments. The top edges are also cleaner due to better alignment of the sides with the top.

The parameters for aligning this model are only slightly different than those used for David's head. Because there are only a tenth as many scans, we used ten times the sampling rate to achieve a similar feature density (0.075%). Also, as discussed in Section 3.1.2, the top surface of these fragments required taking only the highest-stability points for ICP, while the non-pathological edges and bottom used the (more robust) stability sampling. Correspondences required 48 CPU hours, while global registration took 27 minutes.

Awakening: This model (Figure 3.9) consists of 1836 range scans. While this is less than 50% more scans than David's head, the model is much larger, with a lower degree of overlap. As with the other models, our non-rigid alignment improves the sharpness while reducing artifacts in areas of high detail such as the face and thigh. Correspondences required 51.5 CPU hours, and global alignment required 1 hour.

3.2.2 Other 3-D Scanners

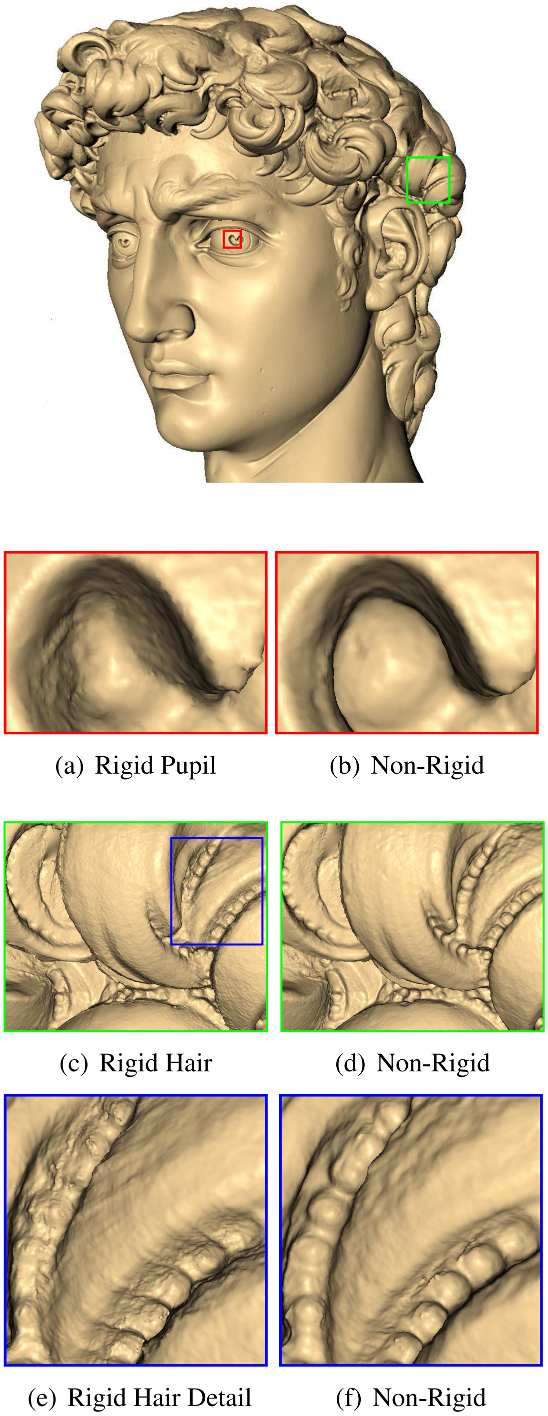
While the Digital Michelangelo scanner achieves high resolution for its working volume, and consequently has especially large deformations, the problem is not limited to this scanner. As shown in figure 3.2, the commercially-available NextEngine Desktop 3D Scanner and Minolta Vivid 910, both based on laser triangulation, contain small yet measurable deformation, and we expect that other commercial scanners do as well. Once again, the non-rigid alignments reduce the alignment error (Figure 3.10). While these deformations are tenths of millimeters rather than millimeters, the working volume is also an order of magnitude smaller.

We have also tested our methods on datasets produced by scanners not based on the principles of triangulation. In particular, the penguin dataset in Figure 3.11 was captured using photometric stereo [Woodham 1980b], which estimates the surface *normals* based on observations of a surface under multiple lighting conditions. These normals are integrated to obtain surface geometry, and any small bias in the normals, such as that due to interreflection, accumulates to produce warped geometry. Here too, non-rigid alignment can help. Figure 3.11 shows the penguin model scanned using photometric stereo and aligned using rigid and non-rigid alignment. While Nehab *et al.* [2005] have demonstrated the ability to correct for this bias by combining photometric normals with triangulation data, our methods can work with the biased normal scans directly.

The penguin required 25 minutes for correspondence computation and one minute for global alignment, while the warrior required 1.5 hours for correspondences and 2 minutes for global registration. Because these models are so much smaller, the feature sampling rate was increased to 0.1%, the minimum distance between features was reduced to 3 mm, and the minimum stability threshold for correspondences was raised to 0.01.

3.3 Discussion

We have presented an algorithm for global non-rigid alignment of three dimensional range scans. Our algorithm is robust not only to noise, but also to non-linear warp caused by scanner calibration error. We recover accurate feature correspondences using a locally weighted, stable ICP computation, then obtain a consistent set of range scans using thin-plate spline warps. The resulting merged models contain more detail than rigidly-aligned versions when scans contain warp. The algorithm scales well with data size, requiring running time proportional to the number of overlapping pairs of scans. Because of its

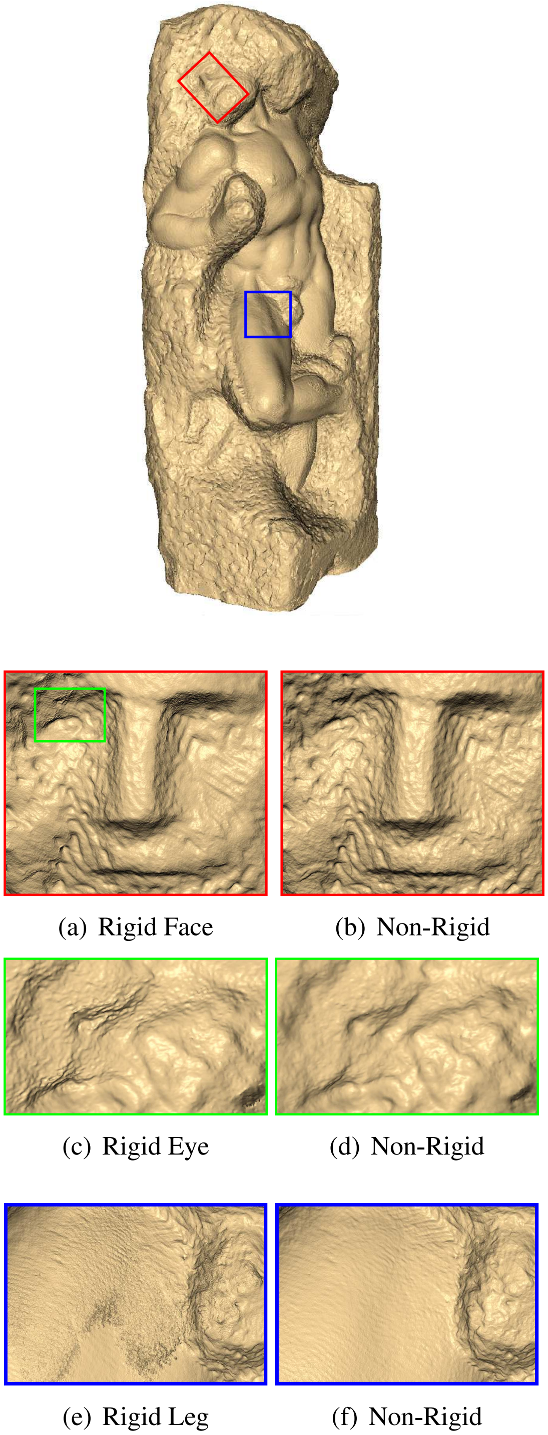


(a) Rigid Pupil (b) Non-Rigid

(c) Rigid Hair (d) Non-Rigid

(e) Rigid Hair Detail (f) Non-Rigid

Figure 3.8: Comparisons of rigid and non-rigid alignments of David's head. The latter are sharper, with fewer merging artifacts. Note the distorted shape and edges of the pupil (Figure (a)) and artifacts in the chisel marks in the hair (Figure (e)).



(a) Rigid Face (b) Non-Rigid

(c) Rigid Eye (d) Non-Rigid

(e) Rigid Leg (f) Non-Rigid

Figure 3.9: Michelangelo's Awakening. Non-rigid alignment yields sharper detail, especially on the face, while reducing misalignment artifacts such as on the leg.

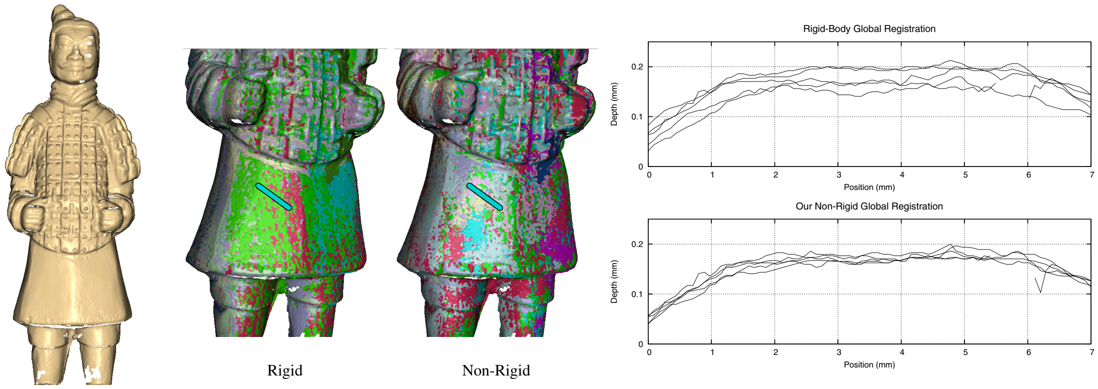


Figure 3.10: The warrior was scanned using the commercially-available NextEngine 3D Desktop Scanner — a laser triangulation scanner. The central images show each range scan in a different color under rigid and non-rigid alignment. The large fields of a single color in the rigid alignment indicate biased misalignment due to warp; the more mottled color of the non-rigid alignment shows that this bias has been removed. The plots at the right show the depths of each range scan along the marked line.

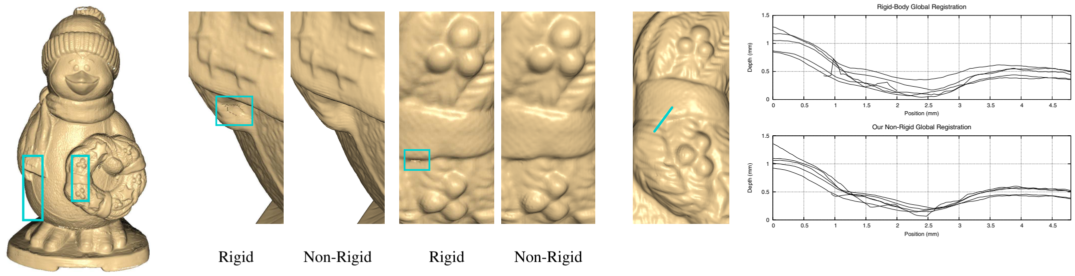


Figure 3.11: This penguin was acquired using photometric stereo, which acquires and integrates a normal field rather than capturing geometry directly. Small errors and bias in the normal field (due to interreflection, for instance) compound to produce large warps in the geometry. Under rigid-body alignment, these lead to increased blurring and to "tears," two of which are shown here. The depth plots at the right show all range scans under rigid and non-rigid alignment along the marked line.

scalability and efficiency, we have found it to be a practical method for rigid as well as nonrigid alignment of models of all sizes.

It is important to be clear about the ways in which our results are "better" than those produced by rigid-body alignment. Indeed, we can not claim to produce results whose overall shape is necessarily closer to the original, since there is just as much uncertainty in the overall position of the scans. However, by avoiding the loss of high-frequency detail caused by low-frequency warps, our final meshes are more *precise* than those produced using rigid-body alignment. Moreover, because of the deformation-minimizing properties of thin-plate splines, our results reflect the minimum non-rigid warp necessary to compensate for deformation in the data. As imagers and 3D scanners become more ubiquitous and inexpensive, and as they are pushed to their resolution and working-volume limits, we believe that in many situations calibration will become the ultimate limit on achievable quality. Our algorithm produces results that maximize the high-frequency fidelity of the result while introducing minimum deviation from the original data.

Chapter 4

Acquisition and Alignment of Fresco Fragments

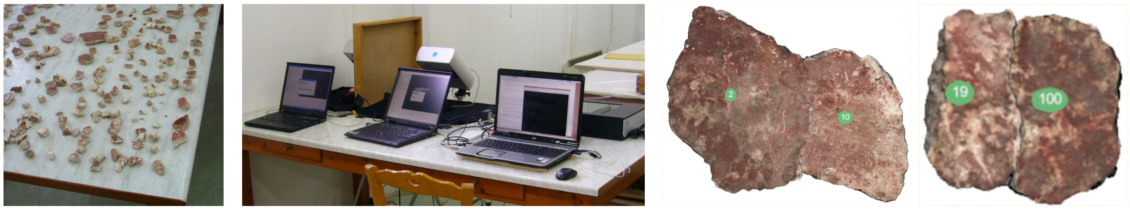


Figure 4.1: Our acquisition system, deployed at the Akrotiri Excavations, Thera. We use a flatbed scanner to capture high-resolution images and normals of wall painting fragments (shown at left), and multiple 3-D scanners to acquire geometry. A single user can operate up to four scanners simultaneously, while a second user operates the flatbed scanner and verifies processing results. This yields a throughput of approximately 10 fragments per hour per user. Our matching algorithm correctly finds the only two matches in this dataset using the scanned 3-D geometry.

We now turn our attention to the problem of scanning many small objects, *e.g.* thousands of fragments from an archaeological excavation. Previous "computational humanities" projects have necessarily involved significant manual labor and nontrivial participation by computer scientists. This is because the tools for high-throughput 3-D scanning, calibrated reflectance measurement, systematic organization of acquired data, and applications such as 3-D shape search have been in the realm of research, rather than production. As these technologies mature, their availability to non-computer experts can be broadened by solving the crucial problems of *scalability* and *usability*. This chapter focuses on the associated technical challenges.

We describe how to integrate acquisition and processing tools recently developed in the computer graphics field into a system suitable for large-scale archaeological documentation and reconstruction. Although we present our work in the context of a case-study, described below, our methods for rapid acquisition by non-experts, automatic and robust 3-D reconstruction, user-focused workflow, and scalable 3-D search are applicable to many archaeological excavations and cultural heritage projects, and could be deployed broadly with minimal incremental development effort.

We focus on the specific problem of documenting and reconstructing fragments of wall paintings from the excavations at Akrotiri. The key challenge to building an effective acquisition system is minimizing manual labor without sacrificing data quality. Analysis tools such as matching algorithms face the same challenge. From this perspective, scalability and usability issues remain with each technology and algorithm suggested above. Therefore, we make the following contributions:

- An inexpensive and rapid 3-D model acquisition system that produces complete models of at least 10 fragments per hour with a single non-expert operator. The most important component is a robust registration algorithm for automatic, high-quality range scan alignment in the presence of unstable or degenerate geometry (*e.g.*, the flat surfaces of wall painting fragments).

- Algorithms for obtaining 2-D color scans and normals (used to document fine relief on the front surfaces of some wall paintings), and registering these to the 3-D geometry. All scanned data and metadata are stored in a database, permitting fragment queries according to a variety of criteria.

- An efficient algorithm for computing candidate 3-D matches between pairs of fragments. In contrast with descriptor-based methods, our algorithm incrementally computes the exact matching error at all possible orientations. Using a novel mesh parameterization, we obtain the performance of a 2-D contour matcher while computing matching error on the full 3-D geometry.

Our approach builds on many previous collaborations between computer graphics, art history, and archaeology. The Digital Michelangelo [Levoy *et al.* 2000], *Forma Urbis Romae* [Koller *et al.* 2006], and Pietà [Bernardini *et al.* 2002] projects, among many others, have demonstrated the usefulness of high-resolution 3-D scanning: the David model was used to plan a cleaning [Bracci *et al.* 2004], several new matches between *Forma Urbis* fragments have been found using the 3-D models [Huang *et al.* 2006; Koller *et al.* 2006], and the Pietà model has been used to analyze the statue's damage and subsequent repair [Wasserman *et al.* 2002]. However, the scanners used in these projects were expensive and unwieldy, and operation and data processing required teams of trained experts.

Other projects have focused on archaeological excavations, especially on reassembling pottery (*e.g.* [Willis 2004]). The principal acquisition goal in these projects has been to recover profile curves — cross-sections of pottery through the axis of rotation — which are important for both matching and archaeological study. This requires less complete data than we acquire, but the challenges of scalability and usability remain. Karasik and Smilansky [2007] discuss these issues, and present a system with which a user can capture 3-D scans of 10–15 pottery fragments (sherds) per hour. (As with our system, multiple operators working together can achieve higher throughput using the same hardware.) However, this system mounts sherds in a frame that obscures parts of their edges. With a similar amount of manual labor, we capture *full* 3-D models, as well as high-resolution images and normals.

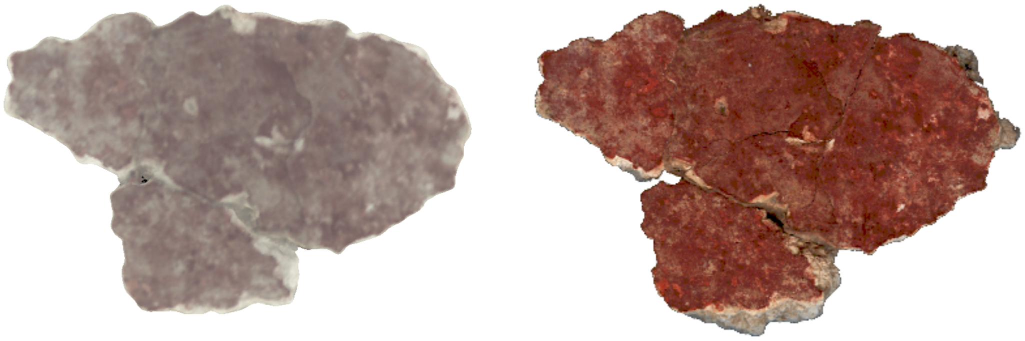


Figure 4.2: Projected 3-D color of the front surface (left) and the corresponding 2-D scan (right). The 2-D scan includes some edge information as well, which is distorted by the flatbed scanner's push-broom projection, directional lighting, and narrow depth-of-field. However the front surface is evenly lit, undistorted, and sharper than the projected 3-D color.

4.1 Design Decisions

Our goal is to design a system and interface for efficient acquisition of geometric and reflectance data for thousands of modest-sized artifacts (5–10 cm across). Because computers and acquisition hardware are inexpensive relative to people, our design decisions favor capturing the most complete representation of an object with the least amount of human supervision. The required time could be further reduced by capturing only the immediately necessary data types, deferring other acquisition until later. But retrieving items from storage is itself time- and space-consuming, and becomes more difficult once fragments have been physically reassembled. We therefore prefer to capture as much data as possible at once.

2-D Images High-resolution images of the fragments' front surfaces are the most basic type of acquired data. The benefits for archiving, analysis, digital restoration, and matching based on decoration are evident. A contour derived from the the front surface image is also useful for matching.

Color images may be obtained with a digital camera, a flatbed scanner, or a 3-D scanner (in conjunction with geometry). We use a flatbed scanner, because it does not require careful control of ambient lighting or fragment positioning, achieves higher resolution than either a camera or 3-D scanner, and guarantees a fixed spatial resolution since the sensor's distance to the object surface is constant. Figure 4.2 compares the projected color obtained from our 3-D scans to a 2-D scan; we believe that the improved color fidelity and resolution justify the additional effort of performing 2-D scans.

Flatbed scanners do have drawbacks. Their depth-of-field is low (only a few millimeters for the hardware we have tested), and we cannot always extract contours for 2-D matching since it is difficult to distinguish pixels belonging to the front side from those coming from the fractured edges. Wall painting fragments, however, have flat front faces (with very few exceptions), we use the scanned color for texture-mapping 3-D models rather than contour matching, and the availability of the 3-D geometry allows us to readily separate the front from the edges, as described in Section 4.3.2.

3-D Geometry While iconographic study of the wall paintings relies primarily on images, geometry is the most important cue for matching. Many matches are found, and all are verified, by testing whether fragments "lock" together. Thickness is used to quickly discard incompatible pairs. The back may contain impressions of material from the underlying mud-brick walls (such as reeds, pebbles, and sand), which are useful cues in the absence of decoration. Finally, contours may be extracted from the intersection of the 3-D model and a plane, which is positioned away from the front and back surfaces to avoid eroded regions.

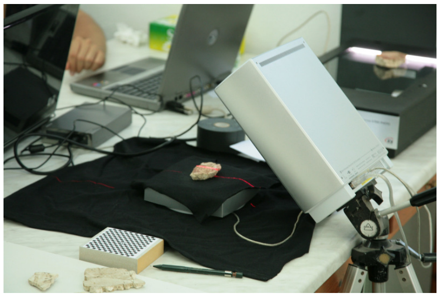


Figure 4.3: Using an off-the-shelf 3-D scanner (front), positioned at a 45-degree angle relative to a motorized turntable, we can acquire the front, back, and sides of a fragment with a single manual flip. A 2-D scanner (rear) simultaneously acquires high-resolution color and normals.

Most fragments are less than 10 cm in diameter, and vary from a few millimeters to a few centimeters in thickness, well within the viewing volume of commercial 3-D laser range scanners. Geometry acquisition with sub-millimeter accuracy is therefore technically feasible: the challenge is minimizing the necessary user interaction. While commercial scanning software such as TowerGraphics Easy3DScan and NextEngine ScanStudio can automatically align a *single* series of scans taken on a turntable, it is impossible to capture all sides of an object in a single scan sequence. Furthermore, wall painting fragments yield flat, degenerate geometry, which standard alignment algorithms cannot handle [Gelfand *et al.* 2003].

By positioning the scanner at a 45-degree angle to a turntable, it is possible to acquire all necessary range scans with *two* scan sequences (Figure 4.1). As shown in Section 4.3.1, this setup also provides enough constraints to align all range scans robustly and automatically, despite degenerate geometry.

Normals In Akrotiri wall paintings, artists frequently used string impressions as an aid to demarcate long, straight color edges [Vlachopoulos 2008]. More intricate designs such as spirals were sometimes prepared with narrow incisions. The impressions are helpful in matching, analysis and restoration, because they survive even where the original pigment does not. Such narrow, shallow markings are difficult to capture with either an image or range scanner. However, because of the sharp change in normal direction, they show up clearly with normal acquisition methods, as seen in Figure 4.8.

Several systems have been proposed for capturing normals, using both calibrated [Woodham 1980b; Bernardini *et al.* 2002; Nehab *et al.* 2005] and uncalibrated [Toler-Franklin *et al.* 2007] point light sources. Instead, we rely on the flatbed scanner's *linear* light source, which is placed at a fixed angle with respect to the moving CCD. Hence, we may compute normals from multiple scans with different object rotations. (We typically use four scans, rotating the object 90° between each.) Combining normal and image acquisition exploits the flatbed scanner's high resolution while keeping the number of discrete scanning steps low.

Our method of obtaining normals is similar to the *linear light source reflectometry* of Gardner et al. [2003]. However, because we obtain a single image per scan, instead of a full time-sequence of images, our setup requires different calibration and normal-computation techniques. These are described in Section 4.3.3.

4.2 Acquisition Workflow

From the user's perspective, there are three discrete acquisition steps: 2-D scanning (images and normals), 3-D scanning, and verification and correction of results. In addition, the system must be periodically calibrated. The manual labor required at each step is largely dependent on interface design. While verification and correction effort is affected by the robustness of our registration algorithms — fewer errors mean fewer corrections — presenting an interface in which corrections are simple and efficient is just as important. We have therefore organized the acquisition and verification workflow into a series of simple operations that can be performed by a single operator in sequence or by several in parallel.

Image and Normal Acquisition The user enters a fragment's ID into the 2-D scanning software, places the fragment face-down on the flatbed scanner, and clicks the "scan" button. The software automatically finds the fragment's location in a pre-scan using background subtraction, scans it, and stores it in the fragment database. The user performs four scans, rotated by approximately 90 degrees, then turns the fragment over and repeats the process.

Acquiring four scans of the front and back surfaces, which is necessary to capture normals, a single user can scan approximately 20 fragments per hour. If normals are not required, only a single scan of each face is performed, for a 70 fragment-per-hour throughput.

Geometry Acquisition The user places the fragment face-down on the turntable, and enters the fragment ID and laser start and end positions. The software acquires a set number of range scans (six by default, but adjustable for large fragments), rotating the turntable between each scan. The user turns the fragment face-up, acquires a second sequence of scans, and moves on to the next fragment. The first fragment is automatically aligned in the background. While scanning is in progress, the user can enter any meta-information and notes (ancient and modern damage, erosion, stains from organic materials, etc., and the excavation finding unit that codes provenance, mode of destruction, and other group-based metadata), although notes can also be entered separately.

The 3-D scanners we use are capable of acquiring approximately five fragments per hour. However, because the process is almost completely automatic, a single user can operate four scanners simultaneously, thereby acquiring 20 fragments per hour.

Maximizing Efficiency We have observed that a number of "tricks" improve the efficiency of acquisition. Although each of these is seemingly trivial in itself, we nevertheless document them here as an aid to practitioners:

- When the flatbed scanner smudges, the operator simply places fragments on a different area of the glass, which reduces cleaning frequency. Our background subtraction algorithm for locating fragments is reliable enough to continue locating the fragment and ignore smudges;

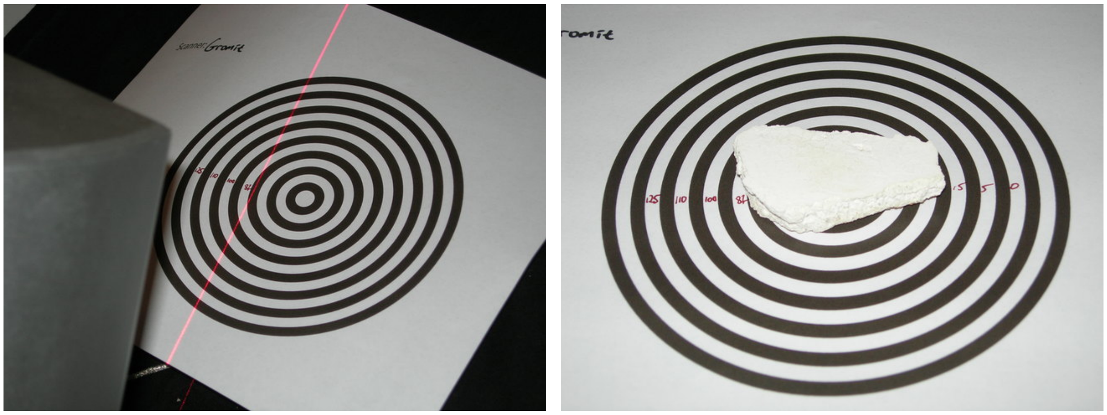


Figure 4.4: To reduce scanning time, a target records scanner settings for several laser positions (left). The operator uses this target to set the laser range for each fragment, so that only the relevant portion of the viewing volume is scanned (right).

- We cover the 3-D scanner's turntable with a black cloth, to exclude its geometry from the scans. Cloth is more diffuse than black paint and dust and debris can be easily shaken off. A small white dot placed at the turntable axis helps center objects;

- The operator uses small wedges, also covered in black cloth, to stabilize fragments that do not sit flat on the turntable;

- A target of concentric circles aids in selecting the laser start and end positions (Figure 4.4). The chart is calibrated before scanning by placing it on the turntable and noting the scanner position when the laser crosses each circle. During scanning, the operator places a fragment on this diagram and reads off the optimal limits for the scan. Selecting tight scanning limits, instead of performing full-range scans, reduces total scanning time (often by a factor of 3) for a typical assortment of fragment sizes;

- The time spent adjusting scan limits is further reduced if the fragments are scanned in approximate order of size;

- Both 2-D and 3-D scanning software play a sound whenever a scan sequence is complete and intervention is required. This allows for easy multi-tasking by operators.

Verification and Correction Once alignment and processing are complete, the user must verify their correctness. First he verifies that the geometry has been correctly acquired and aligned, then that the 2-D image is correctly registered to the geometry. When a processing stage fails, the simplest possible interface is presented to the user to correct *only* that error, after which automatic processing resumes. This minimizes the manual effort spent on each fragment.

There are three possible failure modes for geometry acquisition. If the turntable or scanner is bumped during acquisition, resulting in bad data, the fragment must be rescanned. If the fragment edge is large and flat, it may be incorrectly identified as the front surface; when that happens, the operator clicks on the front surface to automatically reorient the front scans. Finally, if the front scans are misaligned *horizontally* with respect to the back scans, the user repositions them with our constrained alignment interface. Because manipulation occurs within a plane, it is much simpler to control than a full 3-D manipulation interface. With this approach, correct alignments are verified in a second or two, while the occasional error is corrected in a few more seconds. In an experiment involving 150 fragments, verifying and correcting registration required 10 minutes, including correcting all 15 alignment errors.

Problems registering images to geometry are solved similarly to 3-D misalignment: the user provides an approximate initial alignment of the 2-D scan to a *projection* of the 3-D model's front surface, then the software re-optimizes the alignment. Although this optimization is slower than range-scan alignment, the error rate is also lower — only one fragment out of 150 failed to align automatically — so the overall verification time remains small.

Calibration Before acquisition starts, both the flatbed and 3-D scanners must be calibrated. The intrinsic parameters of the 3-D scanner are computed once only, and based on our experience do not change over time. We have also found that professional-quality, flatbed CCD scanners have very nearly square pixels. Any deviation can be compensated for when registering the image and geometry. However, we must calibrate for the 3-D scanner's turntable position (to enable automatic alignment), for the laser positions relative to the turntable (to speed up scanning), and for the flatbed scanner's lighting parameters (for normal acquisition). In practice, we calibrate the flatbed scanner only once, but recalibrate the turntable every day, even when it has not visibly moved.

The turntable position is determined by the plane of its surface, and its axis of rotation. A menu option in the 3-D scanner software guides the user through the calibration process. First, one scan is acquired of the empty turntable to determine its plane. Second, two scans are made of a single fragment, at a 30° rotation from each other. The two scans are automatically aligned (subject to user verification), and the axis of rotation is extracted from this alignment. Note that the axis does not have to be perfect: the multi-way ICP algorithm described below recovers the optimal axis of rotation for each scan sequence using this initial estimate.

Normal acquisition requires knowledge of how the scanner illuminates a diffuse target as a function of angle. We acquire this information by scanning diffuse white cards tilted at angles ranging from 0° to 25° towards the four sides of the scanner.

4.3 Automatic Processing Pipeline

The key to a practical acquisition system is automating 3-D scan alignment and registration of 2-D scans to 3-D models. Failures must be few and easy to correct. Because the classical iterative closest points (ICP) [Besl and McKay 1992; Chen and Medioni 1992; Rusinkiewicz and Levoy 2001] algorithm is unstable in degenerate cases such as a fragment's flat front surface [Gelfand *et al.* 2003], we extend it to operate simultaneously on all scans from a turntable series (*i.e.* all front scans or all back scans). Our extension, called *multi-way ICP*, is restricted to scans on a turntable, but it is faster and more robust than full global registration. We also introduce a novel technique for estimating an initial alignment of the front and back scans to each other without any user input, resulting in a fully automatic alignment of all scans. To register images to the geometry, we optimize for a transformation rather than detecting feature points (*e.g.* [Lowe 2004]) because undecorated fragments do not contain enough meaningful features to compute a robust alignment.

4.3.1 3-D Alignment

Given a turntable's axis of rotation (obtained during calibration), we have good initial estimates of the relative alignments of all front scans and all back scans. We could refine these alignments with ICP, but the nearly-flat front scans often do not contain enough geometric constraints for the optimization to converge. Aligning fronts to backs is even more problematic: we have no initial alignment, since the flip was performed manually. We solve the geometric stability problems within the front and back scans using *multi-way ICP*, and handle the flip by explicitly detecting the front face and using a constrained ICP alignment to robustly align it to the back.

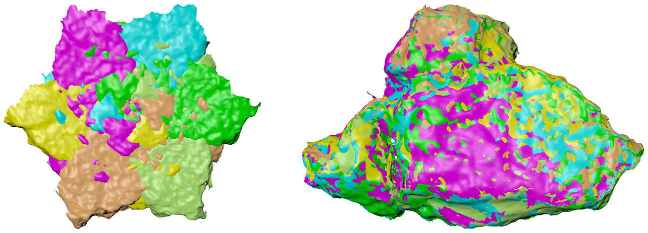


Figure 4.5: Because all six scans were acquired on a turntable, each is rotated 60 degrees with respect to the previous one (left). Using multi-way ICP, we solve for the single rotation that simultaneously aligns all scans (right).

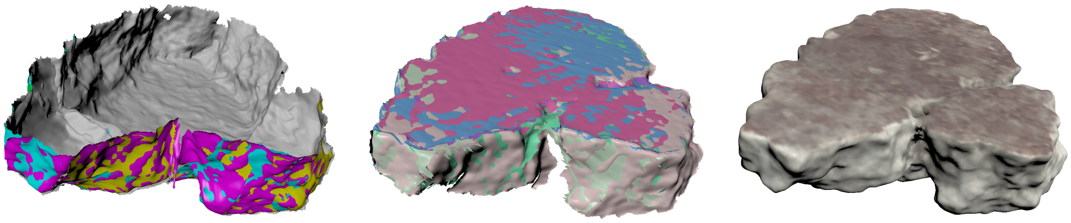


Figure 4.6: Front and back scan sequences and the final model. Because the front and back scans overlap only on the vertical edges, alignment of front to back is unstable in the vertical direction. We compute the vertical alignment directly by detecting the front surface, eliminating the instability.

Multi-Way ICP

For a typical sequence of six fragment scans we have a good initial alignment estimate because they were acquired on a calibrated turntable; any global registration algorithm should align them well. Indeed, even sequentially aligning each scan to the previous one [Chen and Medioni 1992], performs well when all alignments are stable. However, a typical front scan consists of a nearly-flat face, with minimal additional detail from the edge; in this *unstable* situation, each scan can slide parallel to the front surface without significantly increasing the alignment error.

In contrast to the general global registration case, we have important knowledge about the alignment of scans on a turntable. Specifically, for n scans, each scan is rotated $360/n$ degrees with respect to the previous one. This insight leads to the *multi-way ICP* algorithm, which aligns all scans simultaneously with *one* transformation (Figure 4.5).

In multi-way ICP, an initial rotation is estimated based on the number of scans and the turntable's approximate axis of rotation (the exact axis is recovered by the optimization). Corresponding point pairs are selected between each successive pair of scans under the assumption that each scan aligns to the previous one using the estimated rotation. The last scan is assumed to align to the first one, forming a closed loop. At each iteration, a new axis of rotation is computed that minimizes the alignment of all point pairs. Even if each pair of scans is *individually* unstable, we can still extract a rotation from the information distributed among *all* scans because we simultaneously consider point pairs between all successive scans. Using multi-way ICP, we correctly aligned 149 out of 150 fragments; the remaining fragment failed because it moved during scanning.

To solve for a rotation of $360/n$ degrees about *some* axis that minimizes point-to-plane distance between corresponding points, we find an incremental rigid-body transformation, apply it to the current transformation, then coerce the result to be a rotation by $360/n$ degrees. We decompose the transformation into a rotation and translation:

$$\begin{pmatrix} a & d & g & j \\ b & e & h & k \\ c & f & i & l \\ 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{cases} \quad axis = (f-h,\ g-c,\ b-d) \\ angle = \operatorname{atan2}(\|axis\|,\ a+e+i-1)\cdot \\ \qquad \mathbf{t} = (j,\ k,\ l) \end{cases} \tag{4.1}$$

We then set

$$angle' = 2\pi/n, \qquad\qquad \mathbf{t}' = \mathbf{t} - \frac{axis\ (\mathbf{t}\cdot axis)}{\|axis\|^2}, \tag{4.2}$$

and reconstruct the transformation using Rodrigues's formula.

Front-Back Alignment

Unlike alignment of just a front or back sequence, there is no good initial estimate of the relative alignment of fronts to backs. Moreover, there is little overlap between front and back scans — while vertical edges are visible in both sequences, angled edges are only visible in one. Therefore, instead of using a feature-based alignment technique such as spin images [Johnson and Hebert 1997], we directly extract the front surface.

The front surface's flatness, a hindrance for ICP, makes it easy to fit with a plane, using iterative outlier rejection. Of course, the front surface is not visible on the back scans because it is lying flat on the turntable. Since we know the plane of the turntable from calibration, we can rotate the front scans to also lie in this plane, which we define to be the $z = 0$ plane. Any remaining misalignment between front and back scans is an xy planar translation and z rotation. The translation is estimated by aligning centers of mass, and rotation is estimated by trying 40 rotations. From this estimate, we use ICP constrained to a planar transform to refine the alignment (Figure 4.6).

Out of 150 scans, the front surface was incorrectly detected only once. For 13 scans, front-to-back alignment failed. Using our verification interface, each error was corrected in seconds.

4.3.2 2-D to 3-D Registration

Our 3-D scanner provides color registered to each range scan, which we project onto the final model to obtain a colored result. However, the scanner's camera provides only limited color fidelity, lighting is uneven, and resolution is limited. High resolution color obtained under controlled illumination is important for archival purposes, for studying the wall painting's iconography, for matching based on color and texture, and for visually verifying matches.

As noted in Section 4.1, the front surface is difficult to isolate from edge data included in the scanned images. The scanned edges also do not align to a projected image of the 3-D model, making silhouette-based alignments [Lensch *et al.* 2000] unsuitable, because the scanner's depth-of-field is too limited to acquire the entire edge, and the linear light source is highly directional. Feature-based methods are also ill-adapted because the many solid-colored fragments lack stable image features.

Even alignment methods based on color difference or cross correlation are stymied by the extraneous edge data. While we could optimize for the scanner's projection and lighting model, there is a faster and more robust solution: register the scanned image to a projection of *only* the 3-D model's front surface. Recall that the model's front surface is aligned to the $z = 0$ plane, and can be extracted by rendering with the far clipping plane just behind it.

The projected image aligns to a subset of the scanned image (which contains no edges), so we do not penalize scanned image pixels overlapping the projected image's background. In the inverse case, in which a projected image pixel overlaps the scanned image's background, we do assess a penalty. Aside from this, we optimize the alignment using a standard, normalized cross-correlation. To accommodate minor misalignment or misorientation of the 3-D fragment, we solve for an affine transformation (with scale limited to $\pm 2\%$), rather than a rigid one. Figure 4.2 shows a sample result.

To obtain an initial alignment estimate, we use PCA for rotation and center-of-mass for translation. Because 2-D scans include edge data, we have found the estimate is more reliable if we use a projected image that also includes edges for this stage. We improve the initial guess further by trying rotations every 4° within a range of 20° around the estimated PCA axis, and translations every 4 mm up to 20 mm from the center of mass.

Using this procedure we were able to correctly align 2-D scans for 149 of our 150 test fragments automatically. The remaining fragment was corrected by roughly positioning the 2-D scan using our correction interface, and re-optimizing.

4.3.3 Normal Reconstruction

We obtain normal maps via shape from shading, beginning with several (typically four) scans of a fragment at different orientations. Our approach is conceptually straightforward: during a calibration phase we measure $I(n)$, the observed brightness as a function of the surface normal. Then, given a set of scans, we invert I to solve for the normal. As a side effect, we obtain the true color texture (diffuse reflectance) of the surface.

Calibration Because the scanner's light source is linear, not a point, $I(\mathbf{n})$ will not be of the form $\mathbf{n} \cdot \mathbf{l}$. We therefore measure it in a calibration step, using a diffuse plane inclined at known angles with respect to the scanner platform. We measure the intensity at several angles of tilt and rotation to ensure good coverage of the space of normals. Because intensity varies with height off the scan bed, we perform all measurements at a constant height.

We fit a spherical-harmonic model to the diffuse plane measurements to obtain a parametric representation of $I(\mathbf{n})$ that averages out measurement noise and is more easily inverted. Using cross-validation, we have determined that second-order spherical harmonics do not substantially decrease the calibration error over a first-order representation

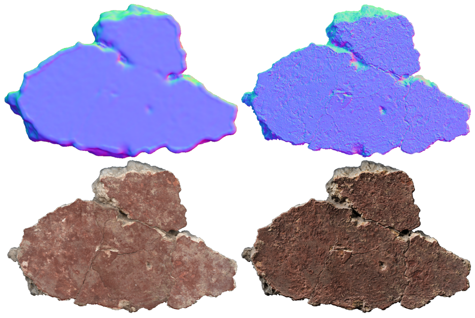


Figure 4.7: Computed normals (top right) reveal more surface detail than those extracted from the geometry (top left). Combining the high-resolution normals with the extracted RGB color (bottom left) allows for enhanced visualizations [Toler-Franklin *et al.* 2007] of surface detail (bottom right).

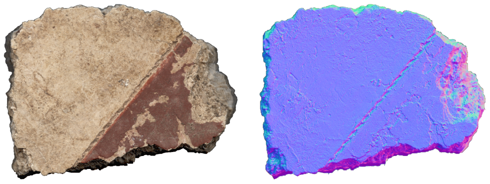


Figure 4.8: String impressions, most clearly visible in the computed normals, mark boundaries of solid color; they are an important cue for reconstruction, restoration, and archaeological study.

containing only constant and linear terms in the normal components:

$$I(\mathbf{n}) = (a_0\, a_1\, a_2\, a_3) \begin{pmatrix} n_x \\ n_y \\ n_z \\ 1 \end{pmatrix} = \mathbf{a}^T \mathbf{n}. \tag{4.3}$$

Normal Computation After capture, we register the scanned images using the algorithm of Section 4.3.2. The inverses of the resulting rotations, which we will call R_i, may be thought of as rotating the light relative to each pixel; in other words, the R_i act on the calibration coefficient vector a. For each image i we may write

$$I_i = \rho (R_i\, \mathbf{a})^T \mathbf{n}, \tag{4.4}$$

where ρ is the diffuse albedo. In matrix form, this expands to

$$\begin{pmatrix} - \; (R_1\, a)^T \; - \\ - \; (R_2\, a)^T \; - \\ \vdots \end{pmatrix} \begin{pmatrix} \rho n_x \\ \rho n_y \\ \rho n_z \\ \rho \end{pmatrix} = \begin{pmatrix} I_1 \\ I_2 \\ \vdots \end{pmatrix}. \tag{4.5}$$

Note that it is not possible to solve this problem as stated using standard linear least squares, as the leftmost matrix in equation 4.5 (let us call it A) is singular: both its last and next-to-last columns will be constant across all rows, because all of the R_i represent planar rotations that leave the z component unchanged. Instead, we define the matrix $\tilde{A}$

to be A with its fourth column removed, giving

$$\tilde{A}\begin{pmatrix}\rho n_x\\ \rho n_y\\ \rho \tilde{n}_z\end{pmatrix} = \begin{pmatrix}I_1\\ I_2\\ \vdots\end{pmatrix}, \tag{4.6}$$

where $\tilde{n}_z = n_z + a_3\,\rho/a_2$. Equation 4.6 can be solved using linear least squares, and by adding the constraint that

$$n_x^2 + n_y^2 + n_z^2 = 1, \tag{4.7}$$

we are able to to solve for ρ, n_x, n_y, and n_z.

Results The normal maps we recover have an order of magnitude higher resolution than our geometry, giving us the ability to document, visualize, and analyze fine surface detail (Figure 4.7), including hairline cracks, string impressions (Figure 4.8), plaster grain, and impressions of reeds or other materials.

Chapter 5

Background: Matching

In the previous chapter, we demonstrated an efficient system for *acquiring* a large number of fresco fragments, however we did not discuss what to do with this data. An obvious application — and one which is essential to restoration and iconographic study — is re-assembly of entire frescoes from the fragments. Conceptually, this amounts to solving an enormous jigsaw puzzle by finding matches between individual pieces. Viewed as a geometry problem, fragment matching is no different from range scan alignment — in either case we wish to minimize the squared distance between surfaces. In the next chapter we will discuss matching algorithms we have devised which compute all possible alignment of each fragment pair, and are directly based on or inspired by ICP. However, it is possible to use higher-level shape features rather than an exhaustive search, and other cues such as color and 2-D contours have also been used for matching. We review several of these methods below.

5.1 Color Matching

With luck, the fragments to be assembled include texture or color information. While any of the geometry-based assembly methods described in the following sections could be augmented to take color along edges or surfaces into account, it would be nice to consider the color information available on the fragment's interior as well.

When matching fragments against a photograph or other ground truth, this reduces to a correlation problem (in which both translation and rotation must be considered). Fornasier and Toniolo [2005] examines the reconstruction of renaissance frescoes in the Chiesa degli Eremitani in Padua, Italy that were destroyed in World War II. Matches are found by computing FFT-based correlations between circular fragment patches and the photographs. Rotation invariance is achieved through the use of circular harmonics. However, the algorithm cannot be used directly to match adjoining fragments, as it relies on overlapping information.

Sağiroğlu and Erçil [2006] extends the idea of correlation-based matching to adjoining fragments by using an inpainting algorithm [Criminisi *et al.* 2003] to extend the decoration past the edge of each fragment. Patches of actual fragment color are then correlated against the inferred extensions. Rotation invariance is achieved using a multi-resolution log-polar method [Wolberg and Zokai 2000]. Image correlation of an image after a log-polar transformation recovers the optimal rotation and scale; transformation is tested in the spatial domain, and accelerated through the use of a multi-resolution image hierarchy.

As we have noted, the Theran frescoes contain large, white regions, where color-based matching will be ineffective. On the other hand, humans are very good at evaluating color cues, so that computer-based matching provides much less advantage over manual methods when decoration is present.

5.2 Jigsaw Puzzle Matching

Some of the earliest work on assembling fragments concerns the assembly of jigsaw puzzles [Freeman and Garder 1964]. Jigsaw puzzles have a number of important constraints which aid matching: they are rectangular, with a flat outside edge, pieces often have exactly four neighbors, and pieces fit together via interlocking "indents" and "outdents". Many puzzles also contain only four-sided pieces. Although there may be measurement noise when digitizing pieces, erosion is not an consideration. These properties lead to a variety domain-specific algorithms that, while instructive, do not apply in a general fragment-matching context.

Freeman and Garder [1964] makes no special assumptions about the type of surface match, except that most matches will be complete matches between two edges (*i.e.* there should be few T-junctions). Each piece boundary is divided into edges at slope discontinuities, and global shape statistics of each edge are used to find candidate matches. Final matching determinations come from measuring the distance between aligned curve candidates. The puzzle is constructed piece-by-piece, spiraling outward from the starting point.

Wolfson *et al.* [1988] solves larger puzzles by assuming four-sided pieces and efficiently computing the alignment error of every pair of sides. Edge pieces are detected separately, and an alignment sequence is sought among all pairwise alignments which result in a closed loop. Then, starting at one corner of the frame and working across the puzzle, all pieces are tested for addition. At all times, the top 200 configurations are maintained, until the puzzle is completely assembled.

A more recent approach [Goldberg *et al.* 2002] eliminates the requirement that pieces have four sides. Instead of explicitly detecting the sides, this algorithm extracts indents, straight edges (to detect edge pieces), and outdents. Ellipses are fit to indents and outdents, and their centers are used as feature points to determine matches. After reconstructing the border, pieces are added to the puzzle which adjoin at least two already-placed pieces. New pieces are added in a greedy fashion, however a global re-alignment is performed on all pieces after each addition to redistribute error.

5.3 Contour Matching

Jigsaw puzzles are special cases of the general contour matching problem. Without distinct edges or features, general contour matchers often take the form of a string matching problem. Kong and Kimia [2001] and Leitão and Stolfi [2002] both represent discretized contours using the curvature at each point. The matching cost between sequences of points on two contours is related to the difference in curvature at corresponding points and difference in length between the two sequences. The optimal matching sequence given a pair of starting points on each contour is found using a dynamic program. To further improve performance, a multiscale approach is taken, in which only the best matches from a low-resolution contour are retained at higher resolutions. Leitão and Stolfi [2002] uses a deep hierarchy and retain the $2^k - 1$ best matches at the k-th level (so that at the end, at most one match remains between each pair of fragments). Kong and Kimia [2001], on the other hand, uses only a two-level hierarchy, but requires a three-way match between contours to prune false positives. This is a very strong constraint, which can be very effective at pruning false positives. However, it also causes false negatives when some fragments are missing from the dataset.

Papaodysseus *et al.* [2002] has previously applied contour matching techniques to Theran frescoes fragments. This work exhaustively searches all possible pairings of fixed-length segments of edges. Alignment is computed based on the first pixel in each segment, and alignment quality is determined by a weighted measurement of both the open area between segments and their intersection area. (Open areas may be caused by erosion, whereas intersection can only legitimately be caused by noise, and must never be large.) A measure of each segment's curvature is used to prune the set of pairs to compare. In contrast, the method we present in the next chapter can, as a special case, compute exhaustive contour alignments asymptotically faster using a convolution-like algorithm. However, we sacrifice the ability to distinguish between open areas and intersection.

One interesting problem that arises in contour matching applications is successful acquisition. Intuitively, a 2-D acquisition system such as a flatbed scanner should be used, but as both Goldberg *et al.* [2002] and Leitão and Stolfi [2002] note, even thin jigsaw puzzle pieces and ceramic fragments cast shadows that degrade the quality of contours. With fresco fragments, which can be 1 cm or more thick, we have also encountered problems with camera projections which further degrade the contour. Our solution has been to extract the contour from the 3-D geometry, at which point we prefer to use all the 3-D information available to us.

5.4 Pottery Re-Assembly

Among the most common finds at excavations are numerous pottery fragments, called sherds. Contour-matching approaches, suitably extended to space curves, are a natural solution to matching sherds. Typically, both torsion and curvature are considered at each contour point [Kong and Kimia 2001; Üçoluk and Toroslu 1999], although Hori *et al.* [1999] uses a P-Fourier descriptor instead. Pottery, however, provides the additional constraint of axial symmetry. Archaeologists commonly document pots based on their axis of rotation and *profile curve* — the pot's cross section in a plane through the axis of rotation — and many computational methods have been developed to extract this information (*e.g.* [Mumford and Cao 2002; Zerroug and Nevatia 1999; Karasik and Smilansky

2007]). Willis [2004] goes further, using the axis of symmetry together with contours to aid in matching. Starting with an initial estimate of the axis, a 6-th order polynomial is fit to the sherd for the profile curve. The axis is re-estimated, and the process iterates until convergence. Then, a probabilistic framework is used to find contour matches which are consistent with the axis of rotation and profile curve, which are re-estimated from all aligned fragments as the pot is reconstructed.

5.5 Surface Matching

All of the matching contexts considered so far are two dimensional (embedded in three dimensions in the case of pottery matching). Full 3-D matching, with six degrees of freedom, is considerably more complex, making an exhaustive search of all possible alignments prohibitive. Two ways to reduce the complexity are grouping surfaces into *facets*, between which to search for matches, and identifying promising features to match.

Papaioannou and Karabassi [2003] takes the former approach. Points on the surface are clustered into facets based on normal compatibility, and into original or fracture surfaces based on roughness. Fracture surfaces are aligned with each other by first aligning their average normals, then using simulated annealing to minimize error. For fracture surfaces which border an original surface, contour matching on the boundary is used to quickly find candidate matches and alignments.

Huang *et al.* [2006] also segments fragments into faces to help restrict matching, but eliminates the requirement that faces completely match each other by computing local features to match instead. Volume descriptors giving the amount of the fragment inside a ball centered at each point, and the distance of each point in the ball from the surface, are computed everywhere. These are related to mean and principal curvature computations, but are more resistant to noise. Features are computed with multiple-size balls, and are clustered by overlapping ranges of value and at multiple scales (so that larger clusters contain smaller ones). Each cluster is then represented by it center of mass, orientation (as defined by PCA), and size and anisotropy signatures computed from its singular values. Similar edge features are computed on face boundaries. Now fine-scale clusters with similar size and anisotropy signatures are considered matches provided all enclosing, larger-scale feature clusters also match. Additional geometric pruning finds sets of feature correspondences that lead to a consistent alignment. Starting from a rough initial alignment of pairs of clusters (using the PCA axes and singular values), surfaces are further aligned using a non-intersecting variant of ICP; any feature correspondence that leads to unstable alignment is rejected. Next, a forward search over all corresponding features between two faces computes the set of candidate alignments (there may be several corresponding to different subsets of correspondences), and a graph-based global optimization over all candidate fragment alignments extracts the final assembly. Final positions are optimized using a global, non-intersecting alignment algorithm.

5.6 General Shape Matching

Shape-based surface matching is a special case of the general shape matching problem, in which the similarity of two models or parts thereof is measured. In this larger context, we may be looking for objects of the same type or rough shape — for example we may wish to query a database of models for anything shaped like a chair — so most algorithms are less concerned with the fine detail involved in surface matching, and more in high-level shape information. In fact, many algorithms compute statistics over the whole object, although many others compute statistics over more limited region in order to handle incomplete objects and finer details. Among the many statistics which have been proposed, perhaps the most simple is the D2 descriptor [Osada *et al.* 2002], which records a histogram of the distances between random pairs of points on the model. The Extended Gaussian Image (EGI) [Horn 1984] histograms the distribution of surface normal directions, and is easily localized to object regions by restricting the region over which the histogram is generated. Still other methods consider an object's symmetries [Kazhdan *et al.* 2002]. Another class of algorithms matches shapes based on images or depth maps generated from many viewpoints (or from a few canonical ones) [Min 2004; Chen *et al.* 2003; Heczko *et al.* 2002].

Note that while most shape descriptors are invariant to translation because they are centered either at a specific point or at the object's center of mass, they are not always invariant to rotation. Among the examples give above, the EGO is a clear example of rotation dependence. Several techniques have been proposed to achieve rotation invariance, including testing all possible rotations [Shum *et al.* 1995], normalizing rotations [Ankerst *et al.* 1999; Podolak *et al.* 2006], sorting histogram bins by value, and by performing a spherical harmonic decomposition of the descriptor values (provided they are parameterized on a sphere, as is the EGI) and discarding the phase [Kazhdan *et al.* 2003].

When descriptors are localized on a model, there is a question of the best way to match them. This leads to a notion of *distinction*, that is evaluating how useful a descriptor center at a particular model point will be when searching for similar objects [Shilane and Funkhouser 2007]. This is similar in spirit to stability sampling for ICP (which selects points for which we can find good correspondences), but is not computed from intrinsic shape data. Instead, distinction measures the query performance of a descriptor against a training database, allowing a context-dependent determination of distinction. For example, the most distinctive feature to identify biplanes in a database of airplanes is the wing. On the other hand, the most distinctive feature for retrieving airplanes from a broader database of objects is the tail.

Chapter 6

Efficient Matching of Fresco Fragments

Returning to the specific problem of matching Theran fresco fragments, we could of course use any of the matching techniques from the previous chapter. However, color matching techniques would be limited to fragments with decoration; contour-matching techniques, while useful, are limited in their ability to match fragments with straight edges; and fresco fragments have very narrow fracture surface, which we expect will limit the effectiveness of feature-based 3-D matching techniques. For these reasons, we have chosen to start with a 3-D shape-matching algorithm adapted to the characteristics of fresco fragments. Like unconstrained 3-D matching algorithms, we use registration error to evaluate matches. This is the digital analog to testing if two fragments "lock" together, which is the way archaeologists and conservators always verify matches. However, in contrast to full 3-D surface matching, we restrict our search to planar transformations, allowing convolution-like optimizations to efficiently try all possible alignments. We can do this because we know each fragment's front surface, which restricts the possible alignments to two dimensions.

A brute-force approach to testing all possible fragment alignments involves selecting a pair of points on the two fragments, computing an initial alignment based on the points and their normals, running ICP (constrained to a planar transform) to recover a candidate matching alignment and error, and repeating the process for points distributed around each fragment. But repeated ICP alignments are slow (requiring, on average, 45 seconds per fragment on our test dataset), and the ill-constrained geometry produces instability in ICP. If many accurate point pairs could be determined very quickly, there would be no need to iterate, and the process would be much faster. Eliminating iteration from the alignment also resolves the stability problems.

Suppose we had a mesh of only a fragment's edges, which we call a "ribbon," and these ribbons had regular sample spacing (Figure 6.2). Each ribbon vertex could be indexed by row and column, so we could select a strip of samples of a fixed width on each fragment, and ask whether they match. Now the correspondences are completely determined by the regular sample structure, and all that remains is to compute the alignment and associated error. The overlap region is then shifted by a single sample, and the process is repeated (Figure 6.1). This is similar to a 2-D convolution, and indeed each new alignment and error can be incrementally computed from the previous one in constant time. The computational cost is therefore $O(nm)$ where n and m are the edge lengths of the two fragments. The ribbon matcher aligns a pair of fragments in an average of two seconds, irrespective of the fragments' thicknesses and of the width of the matching strip.

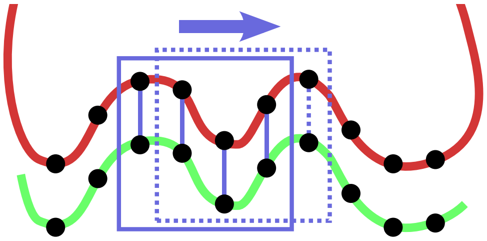


Figure 6.1: A strip of samples on each ribbon is used to compute a candidate alignment. Then, the overlap region is shifted one sample and a new alignment is computed incrementally in constant time.

6.1 Ribbon Construction

We construct ribbons in two stages, as shown in Figure 6.3. First we extract a evenly-sampled contour at a fixed depth, then we extrude the contour vertically along the fragment edge until we reach the top and bottom. For wall painting fragments, we extract the contour 2 mm from the front surface, to avoid erosion and merging artifacts around the front edge. As we explain below, it is important to extract the contour at the same depth on all fragments. We therefore chose 2 mm to accommodate even very thin fragments.

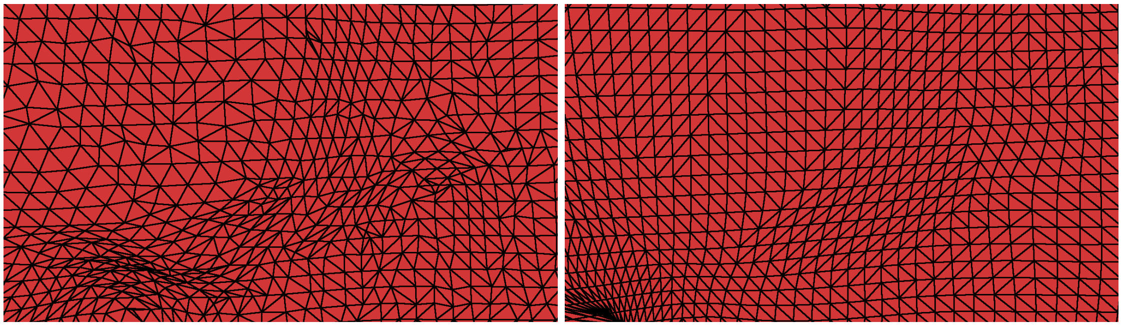


Figure 6.2: Wireframe closeup of the original model and the corresponding regularly sampled ribbon.

6.1.1 Contour Extraction

Conceptually, contour extraction is the process of intersecting a particular plane with the fragment. While algorithms exist to trace the intersection contour of arbitrary, arbitrary-dimensional functions (*e.g.* [Dobkin *et al.* 1990]), we are concerned only with the intersection of a plane and a watertight, triangle mesh that yields a single connected contour. In this case, a simple *ad hoc* algorithm proves effective. To start the extraction, we search for a single edge that crosses the plane, and find its intersection point. Then we walk along the mesh, staying in the prescribed plane, recording a point every 0.25 mm in arclength, until we arrive back at the starting point. This algorithm requires a water-tight, manifold mesh, which is guaranteed by our surface reconstruction. It also breaks down if a mesh vertex lies exactly on the extraction plane, which we handle by perturbing any such vertices slightly beforehand.

If we are to detect a match between two fragments based on point-to-point matching of ribbons, the sample spacing must be dense (because the sample spacing is a lower bound on alignment error), and must be the same between the two fragments. On an erosion-free, noise-free model, sampling by arclength will work perfectly, but either erosion or noise can induce small changes in the contour shape, and therefore in its length. These changes will in turn affect the contour sampling, and harm our ability to detect matches. To provide a measure of immunity to these issues, we smooth the contour points, and reproject each smoothed contour point back onto the fragment mesh along its normal. We use standard Gaussian smoothing, with a standard deviation of 2.5 mm. Experiments with larger smoothing kernels produced identical results.

6.1.2 Extrusion

Next, we walk vertically from each contour point, recording additional points every 0.25 mm until we arrive back at the start point. We call this new contour the *extruded contour* There are two differences from the contour extraction above: we use a vertical extraction plane, and we place points every 0.25 mm by z-coordinate rather than by arclength. Intuitively, this means that if we unroll the ribbon into a flat strip, the points would lie on a regular grid. (In fact, this will only be the case if the edge is exactly vertical. Otherwise stretching will occur along the ribbon.)

The extrusion process extracts many points from the top and bottom surfaces (and the opposite side of the fragment) that are not necessary for matching. To prune these, we walk along the extruded contour from the starting point towards the front surface. The z-coordinate of each contour point will rise until we reach to front surface and start down the other side. We stop as soon as this occurs. Then we repeat the procedure, walking from the starting point towards the bottom surface.

Even after pruning, we are left with many points on the top and bottom surfaces, which are not part of the edge. We eliminate these by triangulating the ribbon (since it is a regularly sampled grid of points, we can use standard range grid triangulation algorithms), and compute connected components of vertices. We discard any component that does not contain a starting point from the original contour.

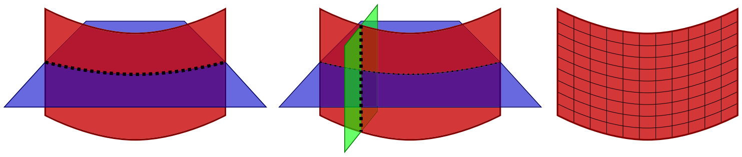


Figure 6.3: **Ribbon Construction:** To efficiently compute fragment pair alignments we regularly resample fragment edges. A contour is extracted at a fixed distance from the front surface, then each contour sample is extruded vertically in a plane defined by the contour point's smoothed normal. In the resampled edge, points are addressable by row and column, allowing efficient correspondence computations.

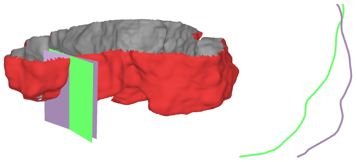


Figure 6.4: A ribbon, with profiles extracted using smooth and unsmoothed normals. We use smoothed normals to ensure consistency between fragments.

6.1.3 Choice of Extrusion Plane

Whereas the original contour is extracted by intersecting a horizontal plane with the fragment mesh at a particular height, extruded contours are extracted by intersecting *some* vertical plane passing through an original contour point. There are infinitely many such planes, and the choice of plane affects the extracted contour (Figure 6.4). If all the extrusion planes for two matching fragments do not line up, their ribbons will be sampled differently and we may not detect the match. We must therefore fix the extrusion plane's orientation in a way that will result in compatible sampling between matching fragment surfaces. We orient the extrusion plane using the smoothed normal at each original contour point, as described in the preceding subsection.

As the fragment edge advances and recedes in the extrusion planes, neighboring *horizontal* samples (*i.e.* samples with the same height, but extruded from neighboring starting points), will spread out or compact together depending on the edge's curvature. So, while each point on the ribbon can be uniquely indexed by row and column, equal arclength sampling holds only for the original contour. This expansion and compression also means the parameterization is dependent on where the original contour was extracted, as well as the choice of extrusion planes. This is why we must fix the original contour extraction at a depth below the front surface which is reasonably safe from erosion, but will still be safely above the back surface for all fragments.

6.2 Incremental Matching with Ribbons

Let us now examine the computational complexity of ribbon matching. Assuming a matching interface w samples wide on a ribbon h samples high, we must consider wh point pairs when computing the optimal alignment and ribbon error at each position. If our fragment contours are l_1 and l_2 samples long, the computational complexity of naive implementation is therefore $O(whl_1l_2)$. Our goal is to reorder the computation in an incremental fashion, thereby reducing the complexity to $O(l_1l_2)$ independent of the strip width or ribbon height.

Given two sets of corresponding points P and Q (the strips from each ribbon), our problem to find the sum of squared distances between them under the optimal rigid-body alignment TR, where T is a translation, and R a rotation, both represented as matrices. Defining the covariance matrix C as

$$C = \sum \tilde{\mathbf{p}}\tilde{\mathbf{q}}^T, \text{ where } \tilde{\mathbf{p}} = \mathbf{p} - \bar{\mathbf{p}},\ \tilde{\mathbf{q}} = \mathbf{q} - \bar{\mathbf{q}}, \tag{6.1}$$

$R = UV^T$ where USV^T is the singular value decomposition of C, and $T = \bar{\mathbf{p}} - R\bar{\mathbf{q}}$ [Arun *et al.* 1987].

In the ribbon matcher context, P and Q are strips from two ribbons A and B. Each time Q is shifted one sample around B, one column of points is discarded from the beginning of Q, and a new column is added to the end. Thus we can reduce the alignment computation's complexity by precomputing column sums. For each corresponding column of A and B, we precompute $\sum p_i q_j$, $\sum \mathbf{p}$, and $\sum \mathbf{q}$. Using the relation

$$\sum(\mathbf{p} - \bar{\mathbf{p}})(\mathbf{q} - \bar{\mathbf{q}}) = \sum \mathbf{pq} - \bar{\mathbf{p}} \sum \mathbf{q} \tag{6.2}$$

(the remaining terms in the expansion cancel, because $\sum \mathbf{p} = \sum \bar{\mathbf{p}}$), we are able to generate the new covariance matrix and alignment from column sums. Furthermore, we can update the covariance matrix incrementally from the column sums as we shift from one column to the next, reducing the total complexity of alignment computations to $O(l_1 l_2)$.

The incremental error computation follows the same principle. Since translation is accounted for by aligning centers of mass, it does not come into play directly in the error computation, and the mean squared error between P and Q is

$$E = \sum ||\tilde{\mathbf{q}}_i - R\tilde{\mathbf{p}}_i||^2 = \sum(||\tilde{\mathbf{q}}_i||^2 + ||R\tilde{\mathbf{p}}_i||^2 - 2\tilde{\mathbf{q}}_i^T R\tilde{\mathbf{p}}_i). \tag{6.3}$$

Because R is a rotation matrix,

$$||R\tilde{\mathbf{p}}_i||^2 = (R\tilde{\mathbf{p}}_i)^T(R\tilde{\mathbf{p}}_i) = \tilde{\mathbf{p}}_i^t R^T R\tilde{\mathbf{p}}_i = \tilde{\mathbf{p}}_i^T \tilde{\mathbf{p}}_i = ||\tilde{\mathbf{p}}_i||^2 \tag{6.4}$$

This gives

$$E = \sum ||\tilde{\mathbf{p}}_i||^2 + \sum ||\tilde{\mathbf{q}}_i||^2 - 2\sum \tilde{\mathbf{q}}_i^T R\tilde{\mathbf{p}}_i. \tag{6.5}$$

For the first two terms, we can once again precompute column sums:

$$\sum ||\tilde{\mathbf{p}}||^2 = \sum(\mathbf{p} - \bar{\mathbf{p}})^2 = \sum \mathbf{p}^2 + 2\bar{\mathbf{p}} \sum \mathbf{p}. \tag{6.6}$$

(This is the identity from Equation 6.2, with $\mathbf{p} = \mathbf{q}$). The third term expands to

$$R_{00}\tilde{p}_x\tilde{q}_x + R_{01}\tilde{p}_y\tilde{q}_x + R_{10}\tilde{p}_y\tilde{q}_x + R_{11}\tilde{p}_y\tilde{q}_y = \mathrm{tr}(RC). \tag{6.7}$$

The individual product terms are just elements from the covariance matrix, which we have already shown how to precompute. Note that any polynomial error metric, including point-to-plane, can be computed incrementally; however, higher-order metrics require more precomputed intermediate terms.

Since both the alignment and error computations require only constant-time update between each ribbon position, the total computation is indeed proportional the the product of the fragment perimeters, or $O(l_1 l_2)$. This is essentially the same optimization used for correlations and convolutions in images, and is possible because we have imposed a similar grid structure on our samples. If we were correlating or convolving entire ribbons against each other, we could doubtless improve performance further using Fourier transforms. However, our problem is instead analogous to correlating many small patches of one image to the other; in this situation Fourier transforms to do not help, and the best performance is obtained in the spatial domain.

6.2.1 Erosion Detection

Many fragments exhibit some erosion, especially near the back surface. In eroded areas, correspondences assigned by the ribbon matcher will be wrong, leading to incorrect alignments (Figure 6.6). ICP eliminates poor correspondences using a normal compatibility constraint— any correspondence in which the normals are not nearly opposite each other is rejected. That constraint requires an

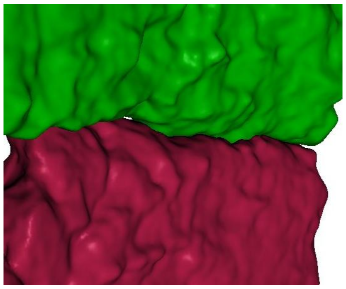


Figure 6.5: Two matching fragments with erosion. To find the correct match, we consider only uneroded points on the ribbon.

approximate initial alignment, but we are able to use a weaker constraint that proves effective. Although a point's normal will be affected by the alignment, it's z-component is constant because the transformation is always planar. Therefore the corresponding points' normals should have opposite z-components even before alignment. Formally, we require that $|n_z + n'_z| <= 0.5$. Significant erosion occurs mostly near the back surface, so normals in eroded areas tend to point down on both fragments. Since they are not opposing, they will be pruned.

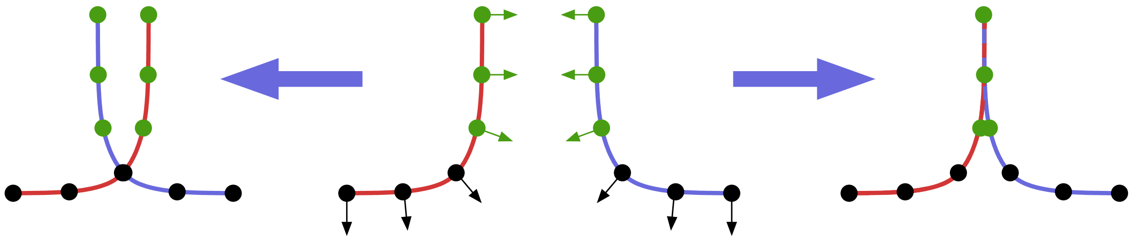


Figure 6.6: Using all corresponding point pairs on the red and blue curves yields an incorrect alignment (left) because the black points are on eroded portions of the edge. Using only the green point correspondences, whose normals' z-components are opposing, yields a correct alignment (right).

6.2.2 Thickness Compatibility

In general, matching fragments are the same thickness. By counting the number of vertical positions along corresponding columns where one ribbon contains a sample but the other does not, we obtain an estimate of the difference in thickness on a per-column basis. We apply a fixed penalty to each unmatched sample beyond a fixed per-column threshold. Typically, we allow a maximum difference of 4 mm (16 samples) in height between fragments, and apply a penalty of 3 mm to each point exceeding the threshold (*i.e.* each such point is counted as having an additional 3 mm of error when determining the final alignment error).

6.3 Selecting Match Candidates

The most natural method to determine a candidate match between a pair of fragments is to take the alignment produce by the ribbon matcher with the lowest error. However, the ribbon matcher error metric does not consider self-intersection, which is an important constraint on matches. Furthermore, since it is more important to reduce the number of false negatives than false positives, we should propose multiple candidate matches for a pair of fragments when one candidate is not significantly better than all others.

To determine the final candidate matches between a pair of fragments, we sort all alignments in increasing order of error. Starting with the lowest-error alignment, we compute the 2-D contour intersection area, and reject the match if it exceeds a threshold (which depends on the properties of the dataset). This prunes candidate matches that do not intersect at the matching interface, but do elsewhere on the fragment. However, area intersection does not distinguish between slight interpenetration along an entire edge, and deep interpenetration. Clearly, the former can occur from imperfection in the data acquisition, whereas the latter definitively excludes the candidate match from consideration.

The 2-D contour also only detects interpenetration at the contour height; when edges profiles are not straight, the 2-D contour can miss interpenetration completely. This is a particular problem on *Forma Urbis Romae* fragments, which are thick, heavily eroded, and contain complex edge profiles. We therefore also compute a 3-D volume intersection, using the corresponding ribbon points along the matching interface. Rather than compute a pure volume intersection, we compute a mean *squared* intersection distance which penalizes deep, narrow interpenetration more than wide, shallow intersection. Because we rely on ribbon correspondences to compute the squared interpenetration, we are restricted to the matching interface (where we have meaningful correspondences), limiting the useful of this heuristic. We therefore record it, for the user to consider when evaluating matches, but do not use it to select or reject candidate matches. In future work, we plan to extend this squared volume intersection to the entire mesh (which can be done efficiently using graphics hardware).

To help avoid false negatives, we include any candidate match passing the contour intersection test whose error is within 20% of the lowest-error candidate match. We enforce a minimum distance constraint between candidate matches — representation transformations as a rotation and translation $(\theta, t_x t_y) \in R^3$, we reject any candidate match whose transformation is with 3 units of an already-considered candidate, under the Euclidean distance metric.

6.4 Matching Results

We have applied our matching algorithm to three test sets of fragments acquired at the Akrotiri excavations from the spirals wall painting and a "ground truth" fresco (which we call the synthetic fresco), all prepared by the conservators to test our system. The test sets contain some known matches found by the conservators, with the possibility that additional matches might exist. To test the generality of our method, we also applied it to a set of fragments from the *Forma Urbis Romae* [Koller *et al.* 2006].

6.4.1 Wall Painting Matching Results

The first test set, of 134 red fresco fragments drawn from the lower, monochrome zone of the composition, contains two matches, both of which were detected by the ribbon matcher (Figure 4.1). The second set, of fragments containing spirals from the main motif, contains 44 fragments and 8 matches, all of which our algorithm finds (Figure

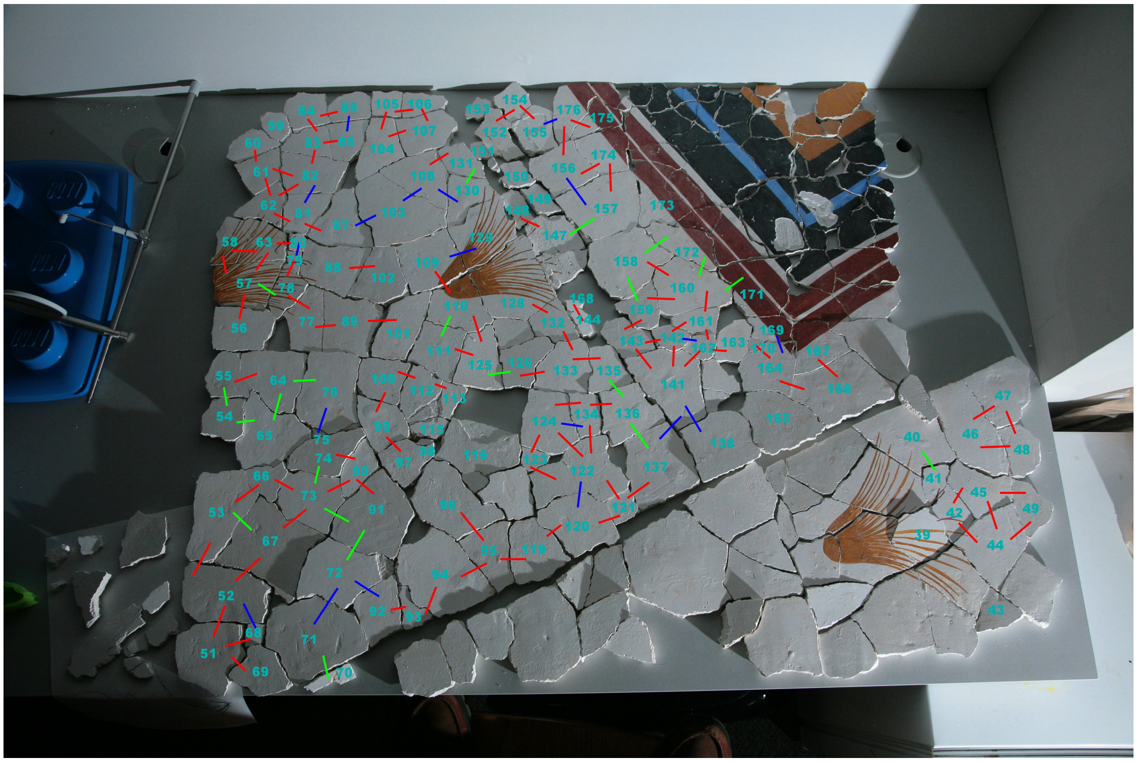


Figure 6.7: Matches found on the synthetic fresco. Red links indicate matches found with a 25 mm strip width. Blue links correspond additional matches found with a 12.5 mm strip width. Green links show matches found only with a 50 mm strip width. Only matches with at least 20% precision are included. 25 mm and 12.5 mm matches were ranked by ribbon matcher error; 50 mm matches were ranked by ICP error on the ribbons. Un-numbered fragments are not included included in the dataset.

6.8). The third set, of white fragments from the unpainted, lime-plaster background, contains 105 fragments. We found one of three known matches in this set, as well as two previously unknown matches. For all three sets, we allowed a maximum 2-D contour intersection of 10 mm^2 when generating candidate matches.

The synthetic fresco contains 129 fragments forming two complete sections of 11 and 118 fragments; as a result there are many matches in this dataset, allowing us to more rigorously test our matching algorithm. We have tested our ribbon matcher with a variety of strip widths, with the best width being 25 mm. However, as we discuss below, different strip widths recover different matches, so we can improve the results further by combining the output of several different ribbon matchers. Longer strip widths tend to

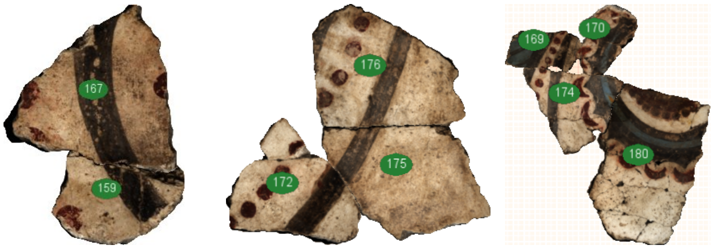


Figure 6.8: The six matches from the spiral dataset, found with the ribbon matcher. The matcher considers only edge geometry; the spiral decoration's continuity is an extra confirmation of success. (We did not find the match between fragments 170 and 174.

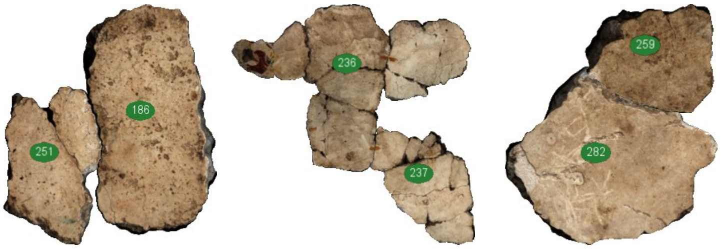


Figure 6.9: Matches from the white dataset.

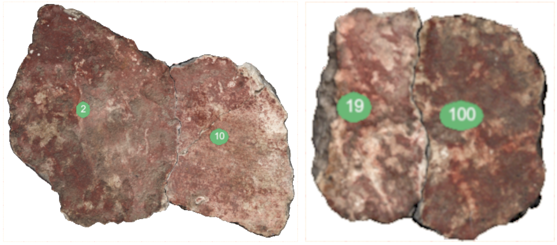


Figure 6.10: Matches from the red dataset.

have higher errors, so we combine results by rank. Results for the different parameter settings are summarized in Table 6.14. Since matches are only meaningful if a human can find them in the ranked list of matches, we plot precision/recall curves for each set of parameters in Figure 6.11. Recall refers to the number of correct matches, and precision to the percentage of correct matches. That is, if the third correct match is 10th in a ranked list, the precision associated with a recall of 3 is 0.3. Once again, we found the best strip width to be 25 mm.

Precision/recall is affected by the ordering of correct matches. We therefore sorted the candidate matches produced by the ribbon matcher according to three distance metrics: the alignment error returned by the ribbon matcher, the alignment error reported by ICP

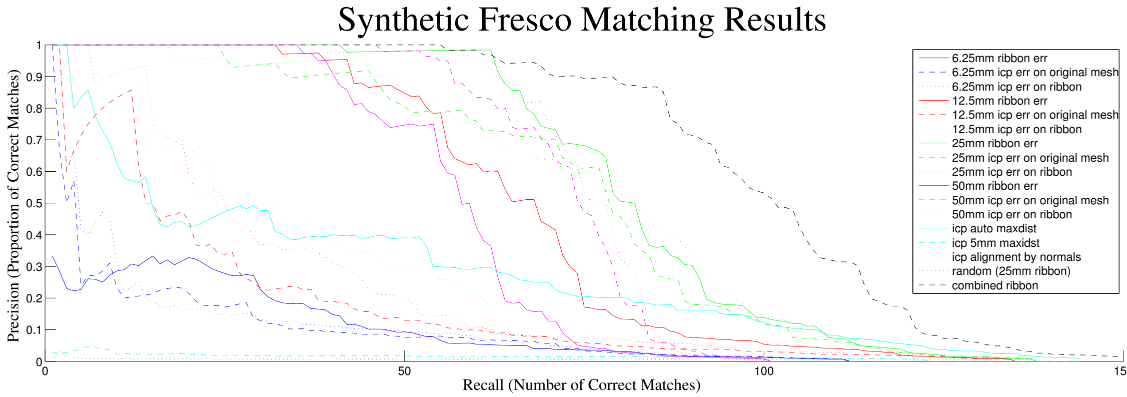


Figure 6.11: Precision/recall of the synthetic fresco matching results. Tests were run of the ribbon matcher using four different strip widths (blue, red, green, and magenta lines), and ranked based on the ribbon matcher's error (solid lines), error computed as in ICP on the original data (dashed lines), and error computed as in ICP on the ribbons (dotted lines). Tests of brute-force ICP matching were performed (cyan) using initial alignments computed using smoothed point normals (dotted) and ribbon matcher output (solid and dashed). For the ribbon matcher alignments, ICP was allowed to automatically set an initial maximum distance threshold for point correspondences (solid) for one test, while it was fixed at 5 mm for the other (dashed). The black dotted line shows precision recall for random ordering of the 50 mm dataset, while the black dashed line shows the rank-combined results for 25 mm, 12.5 mm, and 50 mm ribbons. For combined results, the 12.5 mm and 50 mm output is sorted by ribbon matcher error, while the 50 mm output is sorted by ICP on the ribbons; duplicate *incorrect* matchers are preserved, however duplicate *correct* matches are ignored to avoid unfairly inflating the curve. Note that recall numbers are not normalized, and the test set contains 253 matches.

on the original meshes, and the alignment error reported by ICP on the ribbon. Note that we do not allow ICP to iterate in this scenario, we merely measure the error ICP would report had it converged to this point. In general, the ribbon matcher's error metric is consistently more reliable than the ICP error metric (Figure 6.12). However, for a strip width of 50 mm, we see an inversion — the ICP metric is superior (Figure 6.13). We surmise that many fragments have a matching interface shorter than 50 mm; while the ribbon matcher is finding the correct match, it is including non-matching parts of the fragment edges in its error computation. Because the ICP error metric only measures error on the overlapping fragment portions, it more accurately measures the error associated with the match.

It could be that different strip widths recover different matches, in which case it is valuable to run the matcher with multiple parameters. Indeed, the precision/recall of multiple strip widths combined by rank (with duplicate matches removed) shows this to be the case

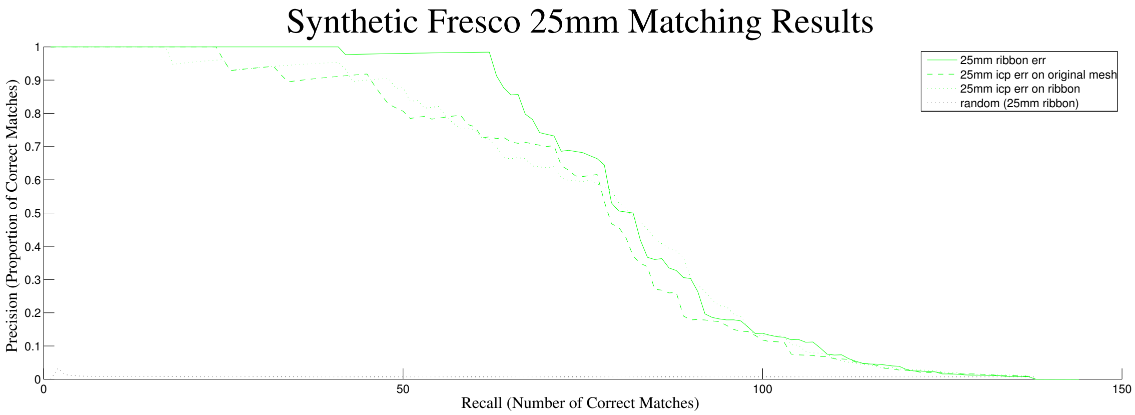


Figure 6.12: Precision/recall of synthetic fresco matching results with the ribbon matcher at 25 mm.

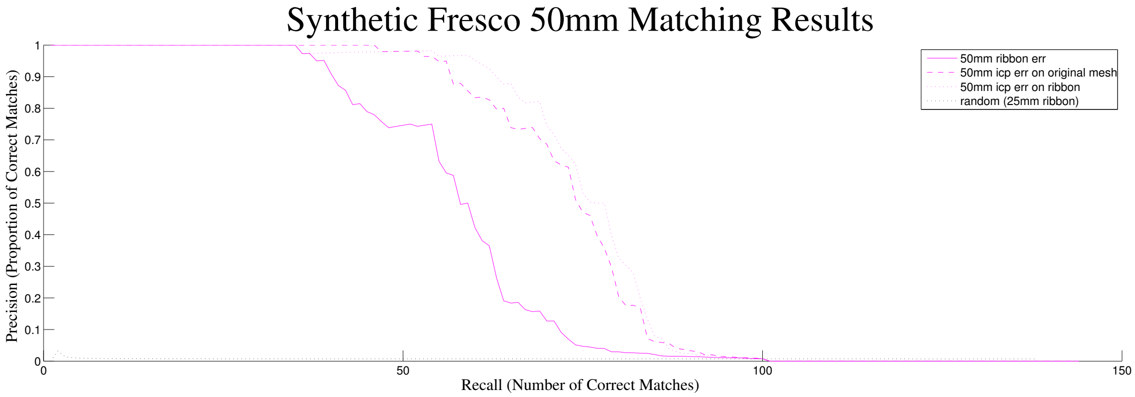


Figure 6.13: Precision/recall of the synthetic fresco matching results with the ribbon matcher at 50 mm.

(Figure 6.15), as does a simple visualization. Figures 6.7 shows the synthetic fresco (only numbered fragments have been scanned), with links showing the matches we found. Red links indicate matches found with a 25 mm strip width (with at least 25% precision), blue links indicate matches with a12.5 mm strip that were *not* found with the 25 mm stripwidth, and blue links indicate matches found with a 50 mm strip width that were not found with either of the two previous settings. (Based on the precision/recall curves, we know that the 25 mm strip width

Matcher and Settings	**Matches**
50 mm Ribbon	101
25 mm Ribbon	138
12.5 mm Ribbon	135
6.25 mm Ribbon	112
Combined Ribbon (12.5 mm, 25 mm, and 50 mm, ignoring duplicates)	175
ICP with alignment by normals	124
ICP with ribbon alignment (auto maximum distance)	145
ICP with ribbon alignment (5 mm maximum distance)	124

Figure 6.14: The number of matches found with the ribbon and ICP matchers with various settings. The test set contains 253 total matches.

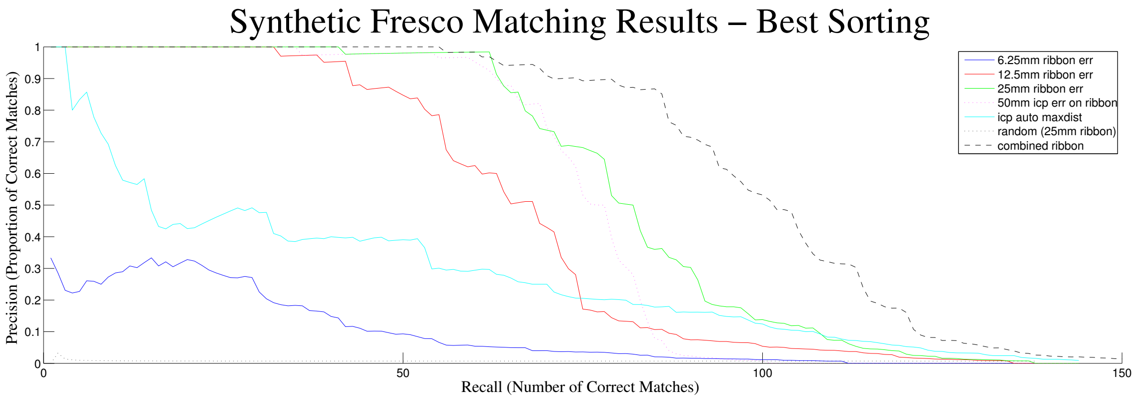


Figure 6.15: Precision/recall of the synthetic fresco matching results using ICP and the best sorting for each ribbon matcher strip width.

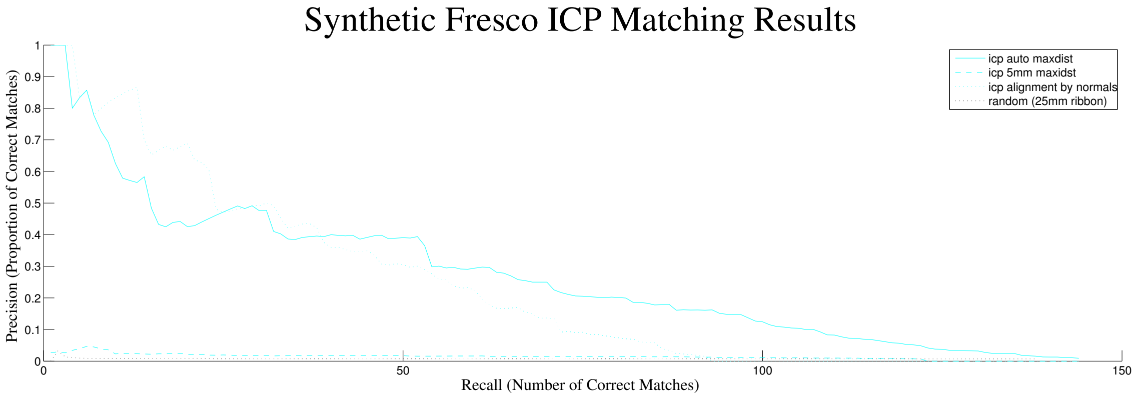


Figure 6.16: Precision/recall of the synthetic fresco matching results with different ICP variants.

finds matches the other parameter settings do not.) Matches are ranked by ribbon matcher error for the 25 mm and 12.5 mm matches, and by ICP error on the ribbons for the 50 mm error. Matching at 6.25 mm finds only one additional match, between fragments 147 and 149, as well as far worse precision/recall performance. We therefore conclude that a longer matching interface is necessary to obtain useful results.

We also tried three variants of brute-force ICP matching, to compare performance and results (Figure 6.16). For the first ICP test, we allowed the ICP implementation to determine an initial maximum distance threshold for corresponding point pairs, and we estimated initial alignments using random points and their smoothed normals; for the second ICP test we initialized the alignments using our ribbon matcher; for the third test

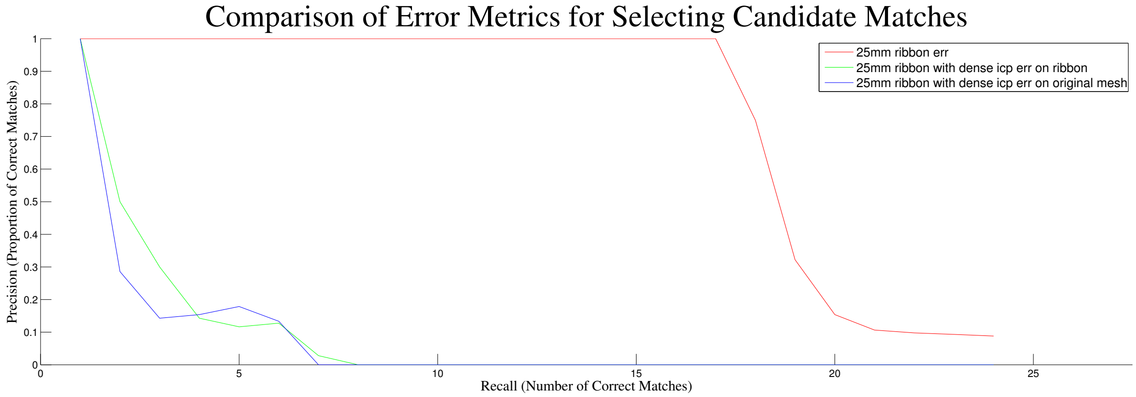


Figure 6.17: Precision/recall of a restricted set of 20 fragments from the synthetic fresco, using different error metrics. The standard ribbon matcher is shown in red. The green and blue lines compute all possible alignments between a pair of fragments using the ribbon matcher, but use the ICP error on the matching interfaces. Unlike the experiments in Figure 6.17, the ICP error is computed at *every* possible alignment, and is evaluated only on the matching interface, rather than the entire mesh. The result of using ICP error on the ribbon meshes is shown in green, and on the corresponding sections of the original meshes in blue.

we continued to use the ribbon matcher, but set the initial maximum distance constraint to 5 mm. While we used initial starting positions for ICP approximately every 20 mm around the edge of each fragment for the first two tests, we increased the frequency to every 10 mm because lower the maximum distance constraint reduces the basin of attraction for convergence. Note that even with these sparse starting positions, the ribbon matcher is nearly two orders of magnitude faster than the ICP matcher. Overall, ICP finds nearly as many matches as the ribbon matcher, but its precision/recall scores are significantly worse, making it far less practical. Although ICP with alignment based on normals has higher precision for early matches than other ICP matchers, we believe the much greater number of matches found at 20% precision using ribbon-based initialization is of more practical benefit. Somewhat surprisingly, setting the initial maximum distance to 5 mm results is much poorer precision/recall performance (although not fewer matches) than using an automatic maximum distance constraint — the precision/recall performance is little better than random. Further experimentation has shown that the problem is one of initial alignment: with only sparse initial alignments, distant point pairs are necessary in early ICP iterations to pull matching fragments toward the correct alignment.

Ideally, we should address the initial misalignment problem in ICP matching by using far more starting positions. This will not necessarily perform better however, because more starting positions will result in more false positives. However, increasing the number starting positions and decreasing the maximum distance constraint naturally leads towards a framework where the ICP error is calculated at *every* ribbon matcher alignment, but without iteration (since one of these alignment should be correct). This amounts to substituting ICP's error metric (evaluated at the matching interface) for the ribbon matcher's, and allows us to evaluate the relative performance of the two metrics, independent of the alignment algorithm. To test the effect of these choice, we evaluated the matching results for a set of 20 fragments (57–62 and 77–89) where candidate alignment errors are computed using the ICP error metric on both the ribbons and the original meshes (Figure 6.17). Due to the enormous computation requirements of this test, it was not feasible include all fragments. We restricted the ICP error to the matching interface on each fragment to directly compare the choice of error metric and effects of mesh resampling. The restricted test set contains 36 matches, of which the ribbon matcher (using 25 mm strips) finds 25. Unsurprisingly, the ribbon error outperforms ICP error for identifying matches within a fragment as well as between them. ICP error on the ribbons finds 8 matches, while using the original meshes it finds only 7. Furthermore, the ribbon error metric finds a strict superset of the ICP error matches; there would be no benefit to combing the different results in this test. Because the ICP error metric cannot be computed incrementally, it was too computationally expensive to run on the entire synthetic fresco. However, the results on a restricted test set already show that its performance is inferior.

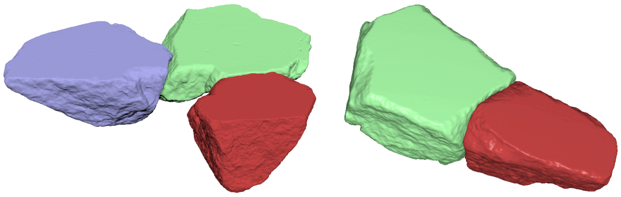


Figure 6.18: Matches in the *Forma Urbis* test set found with our matcher.

6.4.2 Forma Urbis Romae

Our *Forma Urbis* test set contains eight uninscribed fragments, for which edge geometry is the only definitive matching cue. This set contains three representative matches, all of which we found (Figure 6.18). We believe this indicates that our algorithm can be practical in a broad range of contexts.

Forma Urbis fragments are thicker than our wall painting fragments — about 5 cm thick — but because of erosion, matches usually occur along only a small portion of the edge's height. We therefore generated 10 15 mm-high ribbons at different depths along the fragment edge, and matched each set of ribbons independently. Also, because it is impossible to distinguish the front and back surfaces of uninscribed fragments (both sides are perfectly flat), we included each fragment twice, rotated 180° with respect to each other. We used a strip width of 25 mm. Once again, we did not find the 2-D contour intersection test of limited use on this dataset. The reason is not data quality, but that matching at a heigh where erosion has occurred results in interpenetration at a different heigh on the fragments, which is not encoded in the ribbons we generate. To correctly handle intersection on *Forma Urbis* fragments we will need to compute volume intersections on the original fragments or compute contour intersections using the contours from all 10 ribbons. Both of these are areas of future work.

Chapter 7

Conclusion

This thesis has focused on some of the outstanding research problems in 3-D scan alignment for cultural heritage applications. Alignment algorithms have typically assumed well-behaved data, however we have shown that as scanning is pushed to ever higher resolution and (independently) to ever higher throughput, new methods are necessary to address degenerate and warped data. Additionally, large datasets impose new constraints on performance — to be practical, alignment algorithms for large datasets must be parallel and out-of-core, while algorithms for high-volume scanning must be robust and automatic to minimize human effort. Finally, we have discussed the relationship of geometry-based matching to alignment, and shown how to eliminate the performance and stability bottlenecks associated with ICP for matching flat objects. Below, we recall our main contributions, followed by a discussion of directions for future research.

Non-Rigid Alignment of 3-D Scans We accommodate systematic, low frequency range scan bias by incorporating a thin-plate spline warp into a global alignment framework. Our algorithm handles arbitrarily large datasets because it considers only pairs of scans at a time to compute correspondences, making it highly parallelizable and limiting memory requirements. The problem of non-rigid error due to miscalibration is well-documented in the Digital Michelangelo and *Forma Urbis Romae* projects, however we have shown that other, commercial 3-D scanners show a similar amount of warp relative to their viewing volumes. We therefore believe the problem will persist, especially in cultural heritage projects where sub-millimeter detail over very large objects is of great interest.

Robust, Automatic Alignment of Fresco Fragments We have described an acquisition and alignment system for efficient acquisition of fresco fragments at the Akrotiri excavations, where the sheer volume of fragments makes traditional techniques too labor-intensive. Although the flat front surface of fragments makes them difficult to align using standard ICP, we have shown how to exploit it to design an efficient scanning system, a robust alignment algorithm, and simple correction interface.

Efficient, Exhaustive Edge Matcher We have introduced an algorithm to exhaustively test all possible alignments of a pair of fragments in only a few seconds; This allows us to search for matches between fragments based on fine-scale geometric detail rather than on particular, distinctive features. By reparameterizing the edge of each fragment, we are able to eliminate both the correspondence-finding and iteration inherent in ICP. This increases performance by several orders of magnitude, while eliminating stability problems in areas of low or constant curvature. By incrementally updating the alignment, we make the performance independent of the length of the matching interface, yielding asymptotically better performance.

7.1 Non-Rigid Alignment

There are some areas of the non-rigid alignment algorithm which will benefit from further refinement. In certain areas, such as the top of David's head, the acquired range scans are simply of poor quality. Large, flat areas on the sides and bottoms of *Forma Urbis* fragments are fundamentally unstable. By incorporating existing confidence data into our framework, detecting outlier scan regions during the merging process, and dynamically selecting whether to perform locally weighted ICP in unstable regions, we can improve the final mesh quality in these areas. Doing so would likely reduce high frequency error further at the expense of simplicity.

Looking farther afield, we believe our algorithm can be applied to other types of warped data. In particular, image sequences taken with consumer digital cameras contain non-rigid distortions in both space and color due to lack of calibration, as well as perspective warp and parallax. By treating a grayscale image as a height field, it should be possible to sweep most of these distortions under the rug using our (virtually unmodified) algorithm.

Our initial tests of non-rigid image alignment have shown promise (Figure 7.1), but have also revealed particular challenges that must be addressed in the feature correspondence phase. While locally-weighted ICP and stability sampling work well for range scans, images typically contain much sharper edges, requiring even more care in selecting and aligning features. We have therefore experimented with SIFT [Lowe 2004] features, and with refining correspondences by optimizing over a dense patch of pixels (rather than the large regions involved in locally-weighted ICP). However SIFT features and SIFT-based correspondence algorithms only work reliably in the presence of very high frequency information (sharp edges and corners), and do not guarantee an even distribution of features across the image. In future work, we hope to explore feature

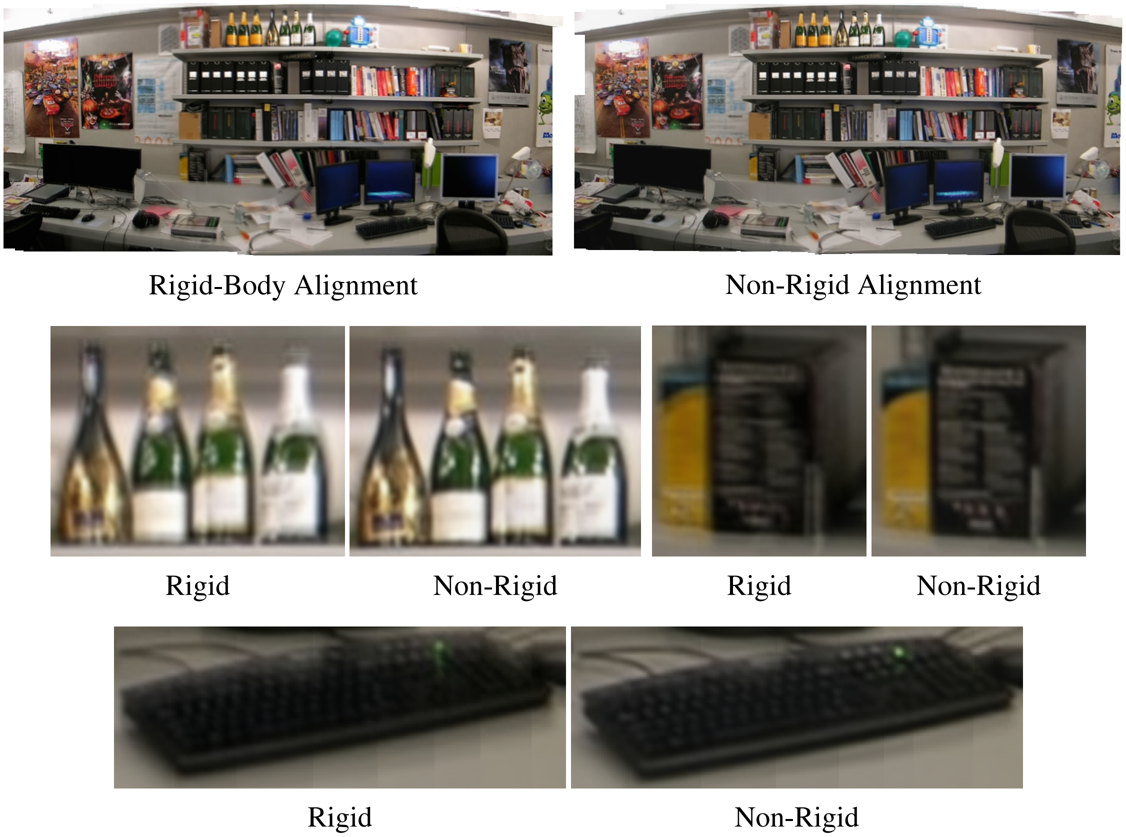


Figure 7.1: Image alignment using rigid-body and non-rigid transforms. Each image in this 28-image sequence was converted to a height field based on intensity values. These were then aligned using a slightly modified version of our 3-D mesh alignment pipeline and averaged together. We have not corrected for autofocus or radial distortion. Closeups show the increased sharpness of the non-rigid panorama due to improved alignment. While we support affine and perspective transforms, they perform worse in this case than the rigid body.

extraction and correspondence finding techniques that are robust across a wide variety of scene types and frequency contents.

3-D scanning remains a data-bound problem. The Digital Michelangelo scans will therefore be eclipsed by even higher-resolution models of even larger objects (such as buildings or entire cities). Any such projects will almost certainly involve a multi-resolution approach to scanning. The ultimate alignment algorithm must therefore efficiently handle multi-resolution scans. At the same time, it should directly incorporate alignment of images, normals, and other data that may not be fully registered registered to the geometry. Current systems must either associate color to individual range scans or register color to a final 3-D model; a truly multi-modal, global, nonrigid alignment algorithm would improve both quality and stability.

7.2 Fresco Acquisition and Alignment

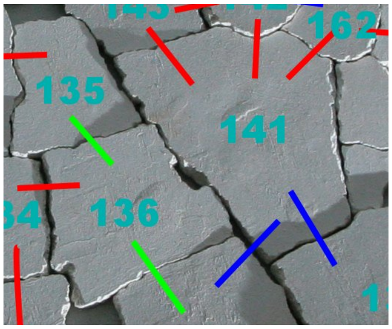
Relative to the effort of excavating, stabilizing, and conserving each fragment, we believe our acquisition speed of roughly 10 fragments per hour is a sufficiently small incremental cost to be practical for documenting *all* excavated items. We have therefore not pursued methods for further automating the scanning process, which would require custom hardware (at significant extra cost) and provide little net benefit. We have also retained the user in the processing loop, seeking only to make it sufficiently robust that the required human time is, on average, seconds per fragment (most of the manual effort arises during acquisition). Nevertheless, hardware improvements could increase scanning efficiency and data quality, particularly with respect to normals. Our current method of extracting normals is limited to the front surface, and image registration remains a challenge; by incorporating a more conventional structured light scanner into the pipeline, we may be able to capture high resolution normals everywhere on the model. Furthermore, if these normals are registered to the 3-D geometry, they can be used to improve model quality as well [Nehab *et al.* 2005].

Generalizability We believe most components of our hardware and processing setup are sufficiently general to be relevant in other applications. The acquisition setup, with a flatbed color scanner and a 3-D scanner oriented at 45 degrees relative to a motorized turntable, is applicable in many contexts, provided that object size is limited to a few tens of centimeters and high-resolution color is required only in flat regions. Our processing pipeline is also generic: multi-way ICP is applicable whenever a turntable is used, as is plane-fitting followed by constrained ICP whenever objects have known-flat regions. While our ribbon-based incremental matcher is most useful for reconstruction of fractured flat surfaces, wall paintings and mosaics are common finds at archaeological sites, as are incised and carved panels of wood, marble, or other stone material.

Nearly Flat Objects Our system relies on a flat front surface for both 3-D alignment and normal reconstruction; although flat surfaces are common, millimeter-scale deviations from the surface plane are also common. The plane-fitting required for 3-D alignment is robust to these deviations, but they result in small image misalignments, which show up as texture embossing when we compute normals. Given the detail present in our 2-D scans, we believe a non-rigid image registration or a shape-from-shading reconstruction should be possible and an interesting alternative to conventional photometric scanning, and is an interesting area for future work.

7.3 Matching Fresco Fragments

Our ribbon matcher would benefit from improved result ordering. At the moment, we rank matches only relative to single error metrics. We should be able to improve the ranking by properly combing multiple metrics into a single number. We also plan to explore additional metrics, such as a distinctiveness score (similar to those used in protein matching), which considers the error of a match relative to other potential matches of that section of fragment edge. Such a metric would help distinguish true matches from flat areas that match everything. Finally, We can reduce false positives by interactively updating the list — as matches are confirmed, all conflicting matches can be removed from the list of candidates.



A match with a gap

There are also several improvements we will explore to our method of determine the best candidate matches for each pair of fragments, especially in the case of short matching interfaces. These should only be considered if the actual matching interface is also short — if we're looking for a 1 mm matching interface, and find a 5 mm interface, we should recover the same match by searching for a 5 mm interface; otherwise it can be considered a false positive. We could also take a dynamic programming approach to finding longer matches: by looking at the matrix of all possible alignments of two fragments, we can detect a long interface by finding a sequence of successive ribbon positions that yield similar alignments with low error. Such an approach would have the additional advantage over a long matching interface that it could allow gaps (due to erosion for example). The match between fragments 136 and 141 of our test fresco (inset) exhibits just such a gap.

The data acquired by our system naturally lend themselves to multi-cue matching. Although we have begun with 3-D shape matching, we believe more relevant matches could be proposed to archaeologists and conservators by considering additional cues. Every matching approach discussed in Chapter 5 could — and should — be applied to fresco fragments, especially feature-based shape matching, and color and texture matching (where we consider surface details such as incisions and string impressions to be texture). Annotations provided by archaeologists also provide important cues which should be incorporated. More importantly, the results of all matchers should be combined in a probabilistic framework, so their results can reinforce each other.

Both distinctiveness and interactive match ordering updates are steps toward a truly global matching algorithm, as in Huang *et al.* [2006], where many matches are considered simultaneously for mutual compatibility. In the long term, we believe the ideal matcher will use a global approach, taking the many local matching cues and distinctiveness into account. However, we think the process will necessarily remain interactive. The computer model will never be as accurate as the original fragment, so manual verification will always be desirable. Similarly, any matching algorithm can benefit from interactive updates that confirm or reject matches.

7.4 Future Directions in 3-D Scanning

Research problems in 2-D imaging have been driven by the availability of data. A first phase of research focused on acquisition technologies themselves (chiefly CCDs). With the ability to acquire digital images and explore their possibilities, new research possibilities opened up in automatic image enhancement and compression, and detection of features within single images (*e.g.* detection of edges [Canny 1986] and faces [Turk and Pentland 1991]). These provided the building blocks for easy, routine image capture using point-and-shoot digital cameras, which has in turn opened up new problems in managing and searching large image databases such as Flickr. Recent research has taken advantage of these databases to focus on problems such as determining semantic image content (*e.g.* [Li *et al.* 2007]), automatic scene completion [Hays and Efros 2007], and determining a photograph's location and vantage point [Snavely *et al.* 2006]. As each wave of research has matured, interest has expanded outward to more complex data types, notably video and 3-D data.

It is no surprise that research in 3-D model acquisition mirrors that of 2-D images. Not only is it informed by the experience with images, the same sequence of events applies: first we need a 3-D scanner to build models, then we need to learn how to process the data so acquisition becomes easy; finally we can think about problems involving large quantities of data. This thesis spans problems in the middle category: registration is a low-level processing task that transforms data into a more usable form. However, now that we understand how to build 3-D scanners and process their data, we expect more attention to the broader challenges we discuss in the Theran Fresco Project: How do we make scanning (and, more generally, 3-D modeling) easy and ubiquitous (*e.g.* Modeling by Example [Funkhouser *et al.* 2004])? How do we manage the data that easy, ubiquitous scanning and modeling provide (*e.g.* 3-D model search engines [Min 2004])? What can we learn by analyzing large 3-D datasets (*e.g.* protein docking [Szilagyi *et al.* 2005] and 3-D face recognition [Blanz and Vetter 2003])? And, as the field matures, it is itself spawning interest in new data types, especially 3-D video (*e.g.*, the Z Corp. ZScanner).

Cultural heritage is a fertile domain in which to research these questions of easy acquisition and analysis. Culture heritage objects are often highly detailed, use a great variety of materials, and range in size from tiny fresco fragments to enormous buildings (or even towns), thereby continually pushing the limits of acquisition and processing technologies. The immediate archival value of 3-D scanning, coupled with the possibilities (some of them realized) of virtual restoration, visualization, dissemination, and education, generate immense interest from archaeologists, conservators, curators, and historians, who are consequently prepared to work with computer scientists on developing usable scanning systems. Although a point-and-shoot 3-D scanner will someday permit larger, more varied model database than we will ever obtain from cultural heritage projects, these will be some of the first to develop large databases, and will drive the user-friendly technologies that will someday result in a cheap consumer device. Cultural heritage projects will therefore continue to play an important role in 3-D scanning research and its applications for many years to come.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *ACM Trans. Graphics (Proc. SIGGRAPH) 23*, 3.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graphics (Proc. SIGGRAPH) 22*, 3.

ANKERST, M., KASTENMÜLLER, G., KRIEGEL, H.-P., AND SEIDL, T. 1999. 3D shape histograms for similarity search and classification in spatial databases. In *Proc. SSD*.

ARUN, K. S., HUANG, T. S., AND BLOSTEIN, S. D. 1987. Least-squares fitting of two 3-D point sets. *IEEE Trans. PAMI 9*, 5.

BENJEMAA, R., AND SCHMITT, F. 1998. A solution for the registration of multiple 3d point sets using unit quaternions. In *Proc. ECCV*.

BERGEVIN, R., SOUCY, M., GAGNON, H., AND LAURENDEAU, D. 1996. Towards a general multi-view registration technique. *IEEE Trans. PAMI 18*, 5.

BERNARDINI, F., RUSHMEIER, H., MARTIN, I. M., MITTLEMAN, J., AND TAUBIN, G. 2002. Building a digital model of Michelangelo's Florentine Pietà. *IEEE Computer Graphics and Applications 22*, 1.

BESL, P. J., AND MCKAY, N. D. 1992. A method for registration of 3-D shapes. *IEEE Trans. PAMI 14*, 2.

BLANZ, V., AND VETTER, T. 2003. Face recognition based on fitting a 3d morphable model. *IEEE Trans. PAMI 25*, 9.

BOOKSTEIN, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. PAMI 11*, 6 (June).

BRACCI, S., FALLETTI, F., MATTEINI, M., AND SCOPIGNO, R., Eds. 2004. *Exploring David: Diagnostic Tests and State of Conservation*. Giunti Press.

BROWN, B., TOLER-FRANKLIN, C., NEHAB, D., BURNS, M., DOBKIN, D., VLACHOPOULOS, A., DOUMAS, C., RUSINKIEWICZ, S., AND WEYRICH, T. 2008. A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings. *ACM Trans. Graphics (Proc. SIGGRAPH) 27*, 3.

BROWN, B., AND RUSINKIEWICZ, S. 2004. Non-rigid range-scan alignment using thin-plate splines. In *Proc. 3DPVT*.

BROWN, B., AND RUSINKIEWICZ, S. 2007. Global non-rigid alignment of 3-D scans. *ACM Trans. Graphics (Proc. SIGGRAPH) 26*, 3.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. PAMI 8*, 6.

CHEN, C., HUNG, Y., AND CHENG, J. 1999. RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images. *IEEE Trans. PAMI 21*, 11.

CHEN, D.-Y., OUHYOUNG, M., TIAN, X.-P., AND SHEN, Y.-T. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum*.

CHEN, Y., AND MEDIONI, G. 1992. Object modelling by registration of multiple range images. *Image and Vision Computing 10*, 3.

CHUI, H., AND RANGARAJAN, A. 2003. A new point matching algorithm for non-rigid registration. *CVIU 89*, 2-3 (February-March).

CRIMINISI, A., PEREZ, P., AND TOYAMA, K. 2003. Object removal by exemplar-based inpainting. *Proc. CVPR*.

CURLESS, B., AND LEVOY, M. 1996. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, ACM Press.

DAVIS, J. 1998. Mosaics of scenes with moving objects. In *Proc. CVPR*.

DOBKIN, D. P., WILKS, A. R., LEVY, S. V. F., AND THURSTON, W. P. 1990. Contour tracing by piecewise linear approximations. *ACM Trans. Graphics 9*, 4.

DOUMAS, C. 1992. *The Wall-Paintings of Thera*. Thera Foundation – P. M. Nomikos, Athens.

DOUMAS, C. 1999. *Santorini: A Guide to the Island and its Archaeological Treasures*. Ektodike Athenon S.A.

DUCHON, J. 1977. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, Springer-Verlag, Berlin.

EGGERT, D. W., LORUSSO, A., AND FISHER, R. B. 1997. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl. 9*, 5-6.

FORNASIER, M., AND TONIOLO, D. 2005. Fast, robust and efficient 2d pattern recognition for re-assembling fragmented images. *Pattern Recognition 38*, 11 (Nov).

FREEMAN, H., AND GARDER, L. 1964. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Trans. Electronic Computers 13*, 2 (Apr.).

FROME, A., HUBER, D., KOLLURI, R., BULOW, T., AND MALIK, J. 2004. Recognizing objects in range data using regional point descriptors. In *Proc. ECCV*.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graphics (Proc. SIGGRAPH) 23*, 3.

GARDNER, A., TCHOU, C., HAWKINS, T., AND DEBEVEC, P. 2003. Linear light source reflectometry. *ACM Trans. Graphics (Proc. SIGGRAPH) 22*, 3.

GELFAND, N., IKEMOTO, L., RUSINKIEWICZ, S., AND LEVOY, M. 2003. Geometrically stable sampling for the ICP algorithm. In *Proc. 3DIM*.

GOLDBERG, D., MALON, C., AND BERN, M. 2002. A global approach to automatic solution of jigsaw puzzles. In *Proc. Eighteenth annual symposium on Computational geometry*, ACM, New York, NY, USA.

GUO, H., RANGARAJAN, A., JOSHI, S., AND YOUNES, L. 2004. Non-rigid registration of shapes via diffeomorphic point matching. In *IEEE International Symposium on Biomedical Imaging*.

HÄHNEL, D., THRUN, S., AND BURGARD, W. 2003. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proc. IJCAI*.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. *ACM Trans. Graphics (Proc. SIGGRAPH) 26*, 3.

HECZKO, M., KEIM, D., SAUPE, D., AND VRANIĆ, D. 2002. Methods for similarity search on 3D databases. *Datenbank-Spektrum 2*, 2. (In German).

HORI, K., IMAI, M., AND OGASAWARA, T. 1999. Joint detection for potsherds of broken earthenware. *Proc. CVPR*.

HORN, B. 1984. Extended Gaussian images. *Proc. of the IEEE 72*, 12 (December).

HUANG, Q.-X., FLÖRY, S., GELFAND, N., HOFER, M., AND POTTMANN, H. 2006. Reassembling fractured objects by geometric matching. *ACM Trans. Graphics (Proc. SIGGRAPH) 25*, 3.

HUANG, Q.-X., ADAMS, B., AND WAND, M. 2007. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Proc. SGP*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland.

IKEMOTO, L., GELFAND, N., AND LEVOY, M. 2003. A hierarchical method for aligning warped meshes. In *Proc. 3DIM*.

JIAN, B., AND VEMURI, B. 2005. A robust algorithm for point set registration using mixture of gaussians. In *Proc. ICCV*.

JOHNSON, A., AND HEBERT, M. 1997. Surface registration by matching oriented points. In *Proc. 3DIM*.

KARASIK, A., AND SMILANSKY, U. 2007. 3D scanning technology as a standard archaeological tool for pottery analysis: practice and theory. *Journal of Archaeological Science*.

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FINKELSTEIN, A., AND FUNKHOUSER, T. 2002. A reflective symmetry descriptor. In *European Conference on Computer Vision (ECCV)*.

KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2003. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. SGP*.

KOLLER, D., TRIMBLE, J., NAJBJERG, T., GELFAND, N., AND LEVOY, M. 2006. Fragments of the city: Stanford's digital forma urbis romae project. In *Proc. Third Williams Symposium on Classical Architecture, Journal of Roman Archaeology*, vol. Suppl. 61.

KOLLER, D., AND LEVOY, M. 2005. Computer-aided reconstruction and new matches in the forma urbis romae. In *Proc. Formae Urbis Romae - Nuove Scoperte, Bullettino Della Commissione Archeologica Comunale di Roma*.

KONG, W., AND KIMIA, B. 2001. On solving 2D and 3D puzzles under curve matching. In *Proc. CVPR*.

KRISHNAN, S., LEE, P., MOORE, J., AND VENKATASUBRAMANIAN, S. 2005. Simultaneous registration of multiple 3D point sets via optimization on a manifold. In *Proc. SGP*.

LEITÃO, H. C. G., AND STOLFI, J. 2002. A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Trans. PAMI 24*, 9 (Sept.).

LENSCH, H. P., HEIDRICH, W., AND SEIDEL, H. 2000. Automated texture registration and stitching for real world models. In *Proc. Pacific Graphics*.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The Digital Michelangelo Project: 3-D scanning of large statues. In *Proc. SIGGRAPH*.

LI, L.-J., WANG, G., AND FEI-FEI, L. 2007. OPTIMOL: automatic Object Picture collecTion via Incremental MOdel Learning. In *Proc. CVPR*.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2.

LU, F., AND MILIOS, E. 1997. Globally consistent range scan alignment for environment mapping. *Autonomous Robots 4*, 4.

LUCAS, B., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *DARPA Image Understanding Workshop*.

MIN, P. 2004. *A 3D Model Search Engine*. PhD thesis, Department of Computer Science, Princeton University.

MITRA, N. J., FLÖRY, S., OVSJANIKOV, M., GELFAND, N., GUIBAS, L., AND POTTMANN, H. 2007. Dynamic geometry registration. In *Proc. SGP*.

MUMFORD, D., AND CAO, Y. 2002. Geometric structure estimation of axially symmetric pots from small fragments. In *Proc. Signal Processing, Pattern Recognition, and Applications*.

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. Graphics (Proc. SIGGRAPH) 24*, 3.

NEUGEBAUER, P. J. 1997. Geometrical cloning of 3D objects via simultaneous registration of multiple range images. In *Proc. SMA*.

OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. 2002. Shape distributions. *ACM Trans. Graphics 21*, 4.

PAPAIOANNOU, G., AND KARABASSI, E. A. 2003. On the automatic assemblage of arbitrary broken solid artefacts. *Image and Vision Computing 21*, 5.

PAPAODYSSEUS, C., PANAGOPOULOS, T., EXARHOS, M., TRIANTAFILLOU, C., FRAGOULIS, D., AND DOUMAS, C. 2002. Contour-shape based reconstruction of fragmented, 1600 BC wallpaintings. *IEEE Trans. Signal Processing 50*, 6 (June).

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graphics (Proc. SIGGRAPH) 26*, 3.

POTTMANN, H., HUANG, Q.-X., YANG, Y.-L., AND HU, S.-M. 2006. Geometry and convergence analysis of algorithms for registration of 3D shapes. *International Journal of Computer Vision 67*, 3.

PULLI, K. 1997. *Surface Reconstruction and Display from Range and Color Data*. PhD thesis, University of Washington.

PULLI, K. 1999. Multiview registration for large data sets. In *Proc. 3DIM*.

ROHR, K., STIEHL, H. S., SPRENGEL, R., BEIL, W., BUZUG, T. M., WEESE, J., AND KUHN, M. H. 1996. Point-based elastic registration of medical image data using approximating thin-plate splines. In *Proc. VBC*.

ROHR, K., FORNEFETT, M., AND STIEHL, H. S. 2003. Spline-based elastic image registration: integration of landmark errors and orientation attributes. *CVIU 90*, 2 (May).

RUSINKIEWICZ, S., AND LEVOY, M. 2001. Efficient variants of the ICP algorithm. In *Proc. 3DIM*.

SAĞIROĞLU, M. Ş., AND ERÇIL, A. 2006. A texture based matching approach for automated assembly of puzzles. In *Proc. International Conference on Pattern Recognition*.

SHARP, G. C., LEE, S. W., AND WEHE, D. K. 2002. ICP registration using invariant features. *IEEE Trans. PAMI 24*, 1.

SHARP, G., LEE, S., AND WEHE, D. 2004. Multiview registration of 3D scenes by minimizing error between coordinate frames. *IEEE Trans. PAMI 26*, 8.

SHILANE, P., AND FUNKHOUSER, T. 2007. Distinctive regions of 3D surfaces. *ACM Trans. Graphics 26*, 2.

SHUM, H., HEBERT, M., AND IKEUCHI, K. 1995. On 3D shape synthesis. Tech. Rep. CMU-CS-95-213, Carnegie Mellon, November.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. *ACM Trans. Graphics (Proc. SIGGRAPH) 25*, 3.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proc. SGP*.

SZILAGYI, A., GRIMM, V., ARAKAKI, A., AND SKOLNICK, J. 2005. Prediction of physical protein-protein interactions. *Phys. Biol. 2*.

TOLER-FRANKLIN, C., FINKELSTEIN, A., AND RUSINKIEWICZ, S. 2007. Illustration of complex real-world objects using images with normals. In *Proc. NPAR*.

TURK, M., AND PENTLAND, A. 1991. Face recognition using eigenfaces. In *Proc. CVPR*.

ÜÇOLUK, G., AND TOROSLU, I. 1999. Automatic reconstruction of broken 3-D surface objects. *Computers and Graphics 23*.

VLACHOPOULOS, A. 2008. The wall paintings from the Xeste 3 building at Akrotiri. Towards an interpretation of the iconographic programme. In *A colloqium on the prehistory of the Cyclades, Cambridge*, Horizon, N. Brodie, J. Doole, G. Gavalas, and C. Renfrew, Eds.

WAHBA, G. 1990. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, PA, ch. 2.4.

WASSERMAN, J., CAMIZ, F. T., VERDON, T., AND ROCKWELL, P. 2002. *Michelangelo's Florence Pietà*. Princeton University Press.

WEN, G., ZHU, D., XIA, S., AND WANG, Z. 2005. Total least squares fitting of point sets in m-d. In *Computer Graphics International*.

WILLIAMS, J., AND BENNAMOUN, M. 2000. A multiple view 3D registration algorithm with statistical error modeling. *IEICE Trans. Information and Systems E83-D(8)* (August).

WILLIS, A. 2004. *Stochastic 3D Geometric Models for Classification, Deformation, and Estimation*. PhD thesis, Brown University.

WOLBERG, G., AND ZOKAI, S. 2000. Robust image registration using log-polar transform. In *Proc. ICIP*.

WOLFSON, H., SCHONBERG, E., KALVIN, A., AND LAMDAN, Y. 1988. Solving jigsaw puzzles by computer. *Ann. Oper. Res. 12*, 1-4.

WOODHAM, R. 1980. Photometric method for determining surface orientation from multiple images. *Optical Engineering 19*, 1.

WOODHAM, R. J. 1980. Photometric method for determining surface orientation from multiple images. *Optical Engineering 19*, 1.

ZERROUG, M., AND NEVATIA, R. 1999. Part-based 3d descriptions of complex objects from a single image. *IEEE Trans. PAMI 21*, 9.