A Hierarchical Method for Aligning Warped Meshes

Master’s with Distinction in Research Report

Leslie Ikemoto
Stanford University
June 2003

Abstract

Current alignment algorithms for registering range data captured from a 3D scanner assume that the range data depicts identical geometry taken from different views. However, in the presence of scanner calibration errors, the data will be slightly warped. These warps often cause current alignment algorithms to converge slowly, find the wrong alignment, or even diverge. In this research report, we present a method for aligning warped range data represented by polygon meshes. Our strategy can be characterized as a coarse-to-fine hierarchical approach, where we assume that since the warp is global, we can compensate for it by treating each mesh as a collection of smaller piecewise rigid sections, which can translate and rotate with respect to each other. We split the meshes subject to several constraints, in order to ensure that the resulting sections converge reliably.

Table of Contents

Acknowledgements

The work in this report was a joint effort with Natasha Gelfand and Professor Marc Levoy. My primary advisor was Professor Levoy, and my secondary advisor was Professor Leonidas Guibas. I would like to thank all of them very much.

Many thanks also go to my parents, sister, and friends for the amazing amounts of love and support they give me everyday. *Aloha nui loa.*

Chapter 1

Technique

Calvin: "I wonder why man was put on earth. What's our purpose? Why are we here?"
Hobbes: "Tiger food." *- Bill Waterson*

Optical 3D scanners typically scan an object from only one direction at a time, so multiple scans are acquired in order to capture an object in its entirety. Assuming that the relative pose of these scans are not known with precision (which is typically the case), the scans must be aligned before they can be merged into a single mesh.

A general alignment pipeline consists of two steps. First, pairwise matching is performed, i.e. for every pair of scans that overlap some fraction of their surface, we compute a pairwise registration. Refer to [3] and [4] for two widely used pairwise registration techniques. Global registration is then performed, i.e. we align the set of scans from the pairwise registration results. A number of global registration techniques have been proposed, such as those discussed in [2], [5], [10], and [13].

The ideal registration pipeline would be robust to the errors introduced by 3D scanning devices. One such source of error is noise, and indeed, many previous techniques are robust to moderate amounts of noise. However, another possible source of error is warp. If the geometry of the scanner used to acquire the range scans is even slightly miscalibrated, or becomes miscalibrated as a result of prolonged use in the field, then significant systematic geometric displacement may be present in the range data. Unlike noise, warpage is typically smooth, as well as monotonic (as opposed to oscillatory). We assume in this paper that the form of this warp is not known.

Unfortunately, existing registration techniques fail to align scans in the presence of even slight warping, since they compute only rigid transformations. In particular, pairwise ICP may converge to a local minimum or oscillate between incorrect alignment results. Figure 1 demonstrates the failure of ICP to align two scans captured under poor calibration. Similarly, in global registration, the scans may converge to a local minimum, oscillate, or even in some cases diverge in shape. Figure 2 illustrates the failure of global registration under incorrect calibration. There are several ways one might attempt to solve the warp problem. One technique that can be used prior to acquiring the range scans is to directly calibrate the scanner accurately. However, this is difficult with a large, reconfigurable scanner, especially if it is deployed in the field like the one used in the Digital Michelangelo project [9]. One can alternatively endeavor to learn the warp by self-calibration of the range scans. We attempted this approach, by locating corresponding point-pairs on overlapping pairs of scans, then expressing each of these points in terms of our particular 21-dimensional scanner space. Using a non-linear optimization system, we then attempted to solve for the scanner parameters. However, the degrees of freedom we considered are not fully independent; thus the system is ill-conditioned.

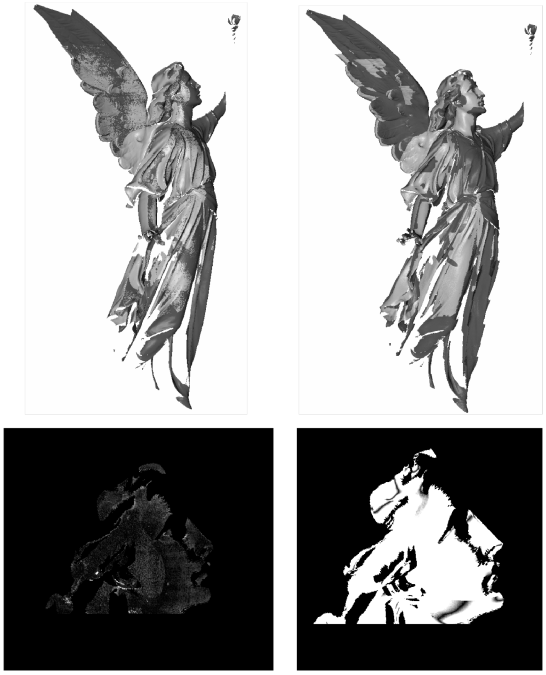

Figure 1. An example of two meshes that exhibit warp. Top left: Two scans acquired under correct calibration; average point-to-plane error over 200 corresponding point-pairs is 0.13 mm. Bottom left: a depth disparity map of the face of the angel (black indicates 0 mm depth disparity between the meshes and white 1.0 mm disparity). Top right: the same 2 scans aligned under simulated incorrect calibration; the average point-to-plane error over 200 point-pairs is 1.81 mm. Bottom right: a depth disparity map of the face. The predominance of white areas shows misalignment on the order of 1.0 mm in many areas. In the Digital Michelangelo Project [9], we often saw this much misalignment due to miscalibration of our scanner.

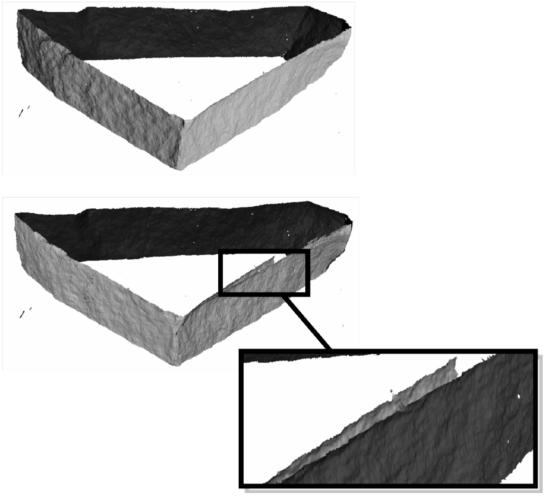

Figure 2. A multi-mesh set exhibiting warp. Top: the scans were acquired under correct calibration, and properly form a closed loop. Bottom: the scans are misaligned due to warp from scanner miscalibration. Inset: 4 mm misalignment between 2 scans (colored different shades of gray). Misalignments during global registration are typically larger than those seen during pairwise alignment because poor pose from even only a single mesh can be propagated to all of its pairwise partners.

Another class of solutions to the warp problem involves introducing a compensating warp into the meshes themselves. There are several options. First, one could fit a low-order polynomial function to the warp. This approach has been proposed for 2D image data in order to produce image mosaics [12]. As a 3D analogue, one can conceive of locally warping overlapping regions of meshes to improve their fit. We are aware of two techniques for fitting a warping function in 3D. Szeliski and Lavallee proposed using an octree-spline to warp a mesh to fit to another [14]. We discuss using this technique as an extension to our method in Chapter 3. We are also aware of work by Bernardini on conformance smoothing [personal communication].

There are two drawbacks to fitting a warping function. First, it introduces many degrees of freedom, and thus may be difficult to control. Second, fitting is likely to be slow, mainly because ICP is an optimized technique for the special case of rigid alignment.

Our solution therefore is to use a piecewise rigid compensating warp, in which each mesh is cut into a set of overlapping regions, which are allowed to translate and rotate relative to each other. The resulting set of regions, each with their own translation and rotation, constitute a piecewise rigid approximation of the (unknown) curved warp. Since this involves solving an ill-conditioned non-linear optimization for which there is no known direct solution, we instead use an iterative solution. An iterative solution adds many degrees-of-freedom, so to improve the robustness and convergence speed of our algorithm, we perform this cutting in a coarse-to-fine manner. While we have used this method only on polygon meshes from range data, it is generally applicable to meshes with arbitrary topology, and to sets of points without connectivity (i.e. point-clouds) as long as normals are defined. Section 1.1 discusses our algorithm in further detail. There are several smaller algorithmic design decisions (e.g., how to subdivide meshes and how to size the overlaps) that we will address further on in Section 1.2.

1.1 Overview of Registration Pipeline

Our strategy for aligning warped meshes is to iterate over aligning the set of meshes using our previous alignment techniques, followed by splitting those meshes that exhibit the worst alignment error into overlapping sub-meshes, thereby allowing them to act independently in the subsequent alignment/dicing iteration. Figure 3 contains the pseudocode for this algorithm.

```
// Given: set of meshes S
// Let I equal set of corresponding points
a. loop until alignment error < $error_{desired}$
b.   for every overlapping P, Q in S,
c.     pairwise_align(P, Q)
d.     I.add(sample(P, Q correspondences))
e.   end for loop
f.   global_registration(S, I)
g.   for (i=0:num_pairs_to_dice)
h.     [$P_{worst}$, $Q_{worst}$] = worst_aligned_pair(S)
i.     [$P_1$, $P_2$] = dice($P_{worst}$)
j.     if (!diced_$P_{worst}$_successfully) break
k.     [$Q_1$, $Q_2$] = dice($Q_{worst}$)
l.     if (!diced_$Q_{worst}$_successfully) break
m.     S.add($P_1$, $P_2$, $Q_1$, $Q_2$)
n.     S.remove(P), S.remove(Q)
o.   end for loop
p.   if no new meshes added to S, break
q. end loop
```

Figure 3. Pseudocode for aligning warped meshes

Alignment (steps (a)-(f)):

We begin each iteration of the loop in (a)-(p) with pairwise aligning every overlapping pair of meshes P and Q in the set of meshes S in steps (b) and (c). The prevailing method used to rigidly align two meshes based on their geometry is the Iterative Closest Point (ICP) algorithm, originally proposed by Besl & McKay [3]. The algorithm finds corresponding points between P and Q, then uses a closed-form method to find a rigid-body transform that minimizes an error metric between the corresponding point-pairs. In each subsequent iteration, the algorithm identities new corresponding points, then recomputes the rigid-body transform. This process continues as long as the error improves. ICP must be initialized with a rough approximation of the true alignment. There are several variants of the ICP algorithm that employ differing error metrics and methods of identifying corresponding points. For a study of these variants, refer to Rusinkiewicz's analysis [11].

In the Digital Michelangelo project (described in Levoy, *et al.* [9]), we used a variant of ICP devised by Chen & Medioni [4]. In this variant, the alignment error is defined as the sum over the set of uniformly sampled corresponding point-pairs of the squared distance from the first point to the plane tangent to a normal defined at its corresponding point. More recently, we devised two improvements on Chen-Medioni. Instead of uniformly sampling points on P and Q, both improvements are non-uniform sampling strategies that sample at a higher rate in areas that exhibit more features, and less in areas that appear planar. The first sampling strategy, termed normal-space sampling, was formulated by Rusinkiewicz in [11]. This algorithm chooses sample points such that their associated normals more uniformly populate the sphere of directions. The second strategy chooses points based on the covariance matrix between P and Q. This algorithm is described in our companion paper [6].

In step (d), a sampled subset of the corresponding point-pairs found in (c) are added to a set of corresponding point-pairs I. Then in (f), we compute a global registration over S, based on this set of point-pairs. Many global registration techniques have been proposed. Bergevin, *et al.* [2] iterates over computing a rigid-body transform for each mesh separately, then applying the set of transforms simultaneously. Stoddart & Hilton [13] attach imaginary springs between the points in each pair of correspondences, and then minimize the energy measure in the resulting dynamic system. Pulli [10] uses pairwise matchings as constraints to incrementally register meshes to a growing set, and includes a backtracking step when the error becomes undesirably large. Fusiello, *et al.* [5] diffuse the error evenly over pairwise rotation and translation results independent of the locations of the corresponding point-pairs, to increase speed of the algorithm.

In the Digital Michelangelo project, we used Pulli's algorithm. Pulli's algorithm attempts to diffuse alignment error evenly over the set of meshes by repeatedly applying Horn's method without recomputing corresponding point-pairs, because doing so is computationally expensive, as Pulli argues.

Since Pulli's algorithm is a least-squares method, the number of corresponding point-pairs we sample and add to I for any given mesh-pair is proportional to how much relative motion between the meshes in the pair will affect the residual error. Thus, the number of point-pairs sampled is inversely proportional to how much the meshes in the pair will be allowed to move relative to each other during global registration. There are a couple of sampling strategies that can be used for these point-pairs. First, we can take into account the same number of point-pairs for every mesh pair, and hence, relative motion in every mesh-pair is penalized evenly. However, we choose not to do this, because we would like to penalize relative motion between mesh-pairs differently depending on how well they form a "lock-and-key" (i.e. how deep and well-defined the global minimum in the error landscape is). We wish to let certain mesh pairs, which do not have many features in the overlap region, slide relative to each other, while we simultaneously constrain the motion of other mesh-pairs which are well-featured in the overlap region. In order to accomplish this, we set the number of point-pairs we add to I for a given mesh-pair proportional to the stability measure for the overlap region given by the method in [6].

Dicing (steps (g)-(o)):

Finally in (g)-(o), we dice some of the meshes in S into halves. We must first decide how many meshes to dice (i.e. how to set `num_pairs_to_dice` in (g)). Dicing a mesh introduces additional degrees of freedom into the system, so the control we have over the compensating warp we are introducing is inversely proportional to the number of meshes we dice. Since we would like as much control as possible, this argues for dicing fewer meshes. However, dicing fewer meshes means we may have to run more iterations of our algorithm, which means repeating the (expensive) rigid alignment steps. Hence, we typically set `num_pairs_to_dice` to one-quarter the number of original meshes.

A second question we must address is which meshes to dice. We could choose meshes based upon known scanner warps. However, one of our assumptions is that the warp is unknown. Another option is to choose meshes based on the modes of deformation of the system. This is an area of future work. Still another option is to use a greedy algorithm to select the pair of meshes exhibiting the worst alignment error. This is what we chose to implement; the meshes are selected in this manner in (h).

Then in (i) and (k), we dice the meshes we selected. Section 1.3 discusses our mesh subdivision algorithm. We add the new meshes to S in (m) and remove the old meshes from S in (n). Then if the alignment error of the system is lower than the alignment error desired, we halt the algorithm. If not, we iterate again.

One caveat to note is that we designed constraints on the geometry and size of the diced sections we create (also discussed in the next section) in order to ensure that the compensating warp introduced in the system is controllable. Thus it may not always be possible to subdivide the particular meshes we chose in step (h) because the mesh is too small or the overlap is unstable. Both are explained in Section 1.3. Steps (j) and (l) check whether the meshes were subdivided successfully. If they were not, we could simply halt the algorithm. However, we may have already added new meshes to S in previous dicing iterations of (g)-(o). Hence, we instead break out of the dicing loop, and check if we added new meshes to S (i.e. if we were able to dice any meshes). If we did, we check the alignment error, and if it is still greater than our desired threshold, we loop back into the rigid-alignment process in (b). If S does not contain any new meshes, we halt the algorithm.

1.1 Hierarchical Splitting of Meshes

In this section, we describe our mesh subdivision algorithm and the constraints we designed to ensure that we can control the compensating warp we introduce.

1.1.1 Subdividing a mesh

To subdivide a mesh M into two regions M_1 and M_2 (as in steps (i) and (k) of the pseudocode in Figure 3), we divide its oriented bounding-box B into two overlapping boxes B_1, B_2 along the longest dimension of B, as shown in Figure 4. B_1 and B_2 will become the oriented bounding-boxes for M_1 and M_2 respectively. The vertices and triangles that B_1 encompasses become M_1, and the vertices and triangles that B_2 encompasses become M_2. In any single iteration of our alignment pipeline, we dice both meshes in a preselected number (`num_pairs_to_dice` in step (g)) of mesh pairs.

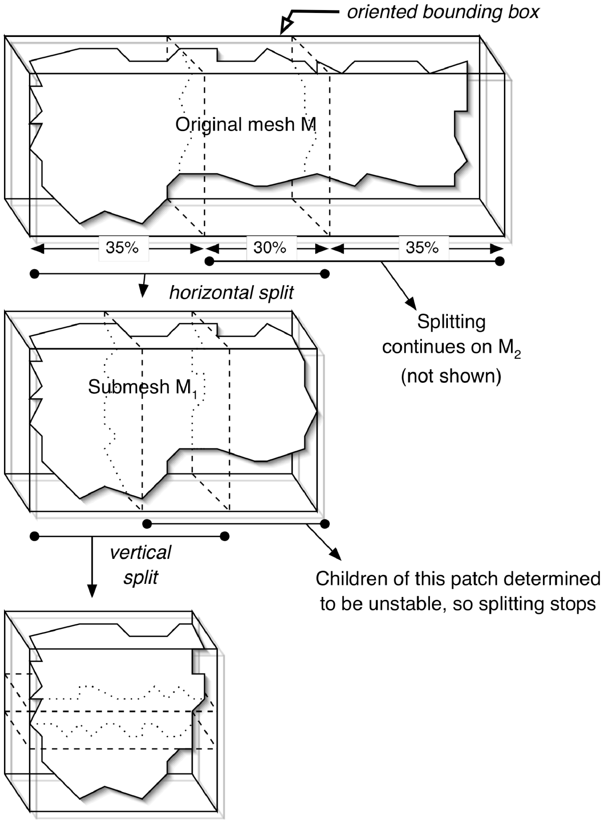
oriented bounding box
Original mesh M
35%
30%
35%
horizontal split
Splitting continues on M_2 (not shown)
Submesh M_1
vertical split
Children of this patch determined to be unstable, so splitting stops

Figure 4. Our subdivision strategy illustrated on a sample mesh M. M is first subdivided along the horizontal direction into M_1 and M_2. Submesh M_1 is then later split horizontally again. The right child is determined to be unstable, so splitting stops. The left child is later split vertically.

Note that Figure 4 is a binary tree, and that the manner in which the tree is traversed determines how each mesh is diced during the course of our algorithm. If we wish to introduce the same number of degrees of freedom to each mesh, we can traverse the tree in a breadth-first manner, in which the original mesh will always be represented by roughly evenly sized regions. Alternatively, if we wish to warp meshes one at a time, we can traverse the tree depth-first so that each region is diced fully before we consider another. We do neither. Instead, we use a greedy algorithm with respect to depth in which meshes are considered equal candidates for dicing regardless of their level on the tree. We conjecture the greedy algorithm will out-perform both breadth- and depth-first traversals. In the pseudocode, no special action is needed for this particular coarse-to-fine strategy.

In order to ensure convergence, we introduce some constraints when subdividing the mesh.

Shape of regions: Scanner warp is difficult to measure, especially in the field (as unfortunately was the case in the Digital Michelangelo project), so we make two simplifying assumptions about the nature of these warps. First, we assume that it is isotropic in all directions. Hence, we dice meshes along their longest oriented bounding-box direction so that in the limit the regions will have an even aspect ratio. Second, we assume that the warp is spatially uniform across the original mesh. Therefore, we are equally willing to subdivide.

Minimum size of regions: The original meshes will (most likely) differ in size. Hence, we control the compensating warp we introduce by the size of each region, not by the number of regions into which we dice each mesh. Every time we dice a mesh, we would like to minimize the number of degrees of freedom introduced, as well as minimize the residual error. Unfortunately, these two goals argue for opposing constraints on the size of the regions. In order to minimize the degrees of freedom, we should create large regions. However, to minimize the residual error, we should create small regions. Fortunately, we already minimize the rate at which we introduce degrees of freedom into the system by taking a hierarchical approach. Thus, we are free to create small regions. But since residual error is proportional to scanner noise, which is proportional to mesh resolution, we need to impose a minimum region size that is proportional to mesh resolution. We currently set the minimum size equal to mesh resolution. In practice, the meshes are usually not (in our experiments) diced that finely because they are not geometrically stable, as explained in [6].

1.1.1 Sizing the overlap

In addition to constraining the minimum region size, we also need to design constraints for creating the overlapping regions within a warped mesh such that their movement in relation to each other during alignment will be well controlled. From a mechanical point of view, we can view the overlap between two meshes as a hinge. The strength of the hinge between two meshes is proportional to the following two quantities.

1. Relative size of overlap region: In order to ensure the compensating warp is controlled, we should avoid excessive motion of the regions relative to each other. As Figure 5 illustrates, corresponding point-pairs between two regions with the same relative rotation will have longer distances and hence a higher squared error if the overlap region is larger. Hence, to prevent excessive relative motion, we should create large overlaps. However, we would like to introduce some warp, which means we need to allow a certain amount of relative motion. This argues for a smaller overlap region. We use an overlap size that is 40% the size of the original meshes (i.e., we set the volume of the overlapping region's oriented bounding box to 40% the volume of B).

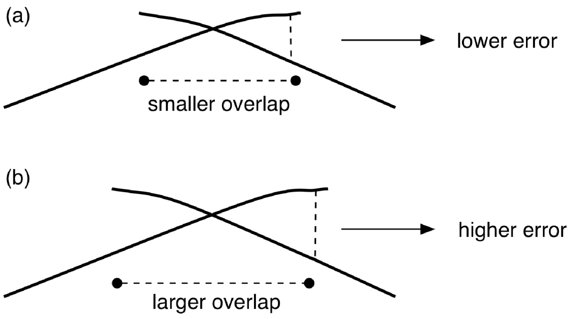

Figure 5. Example showing that overlap size determines the "stiffness" of the hinge between 2 overlapping meshes during global registration. The meshes in (a) and (b) have the same relative rotation, but since the overlap in (b) is larger than in (a), the point-to-plane error in (b) is higher. Thus, the hinge is (b) is stiffer than in (a).

2. Presence of features in the overlap region: Another factor that affects the strength of the hinge between two meshes is the presence and type of features present in the overlap area. There are 3 common types of features we may see in any given area of a mesh. First, the area may be planar, which means our alignment error metric will detect translation errors in z (perpendicular to the plane), but not x-translations, y-translations, or spin about the z-axis. Second, the area may be planar and noisy. This case is even worse than the first, because in the presence of noise the normals may point in slightly incorrect directions, causing the error landscape to have local minima. Third, the area may have stable "lock-and-key" features. With such features, the error landscape for that area will have a deep, well-defined global minimum. Hence, ideally we would like to create overlap regions in areas that have such "lock-and-key" features.

To achieve this, we could analyze the normals of the corresponding point-pairs in a manner similar to normal-space sampling [11]. If the normals sufficiently populate the sphere of directions (specifically, if they create a 3D basis), the mesh pair will be stable during alignment. Alternatively, we could conduct a full stability analysis by analyzing the eigenvalues of the covariance matrix between the two meshes. This idea is explored in our companion paper [6]. At present, we use a full stability analysis, in which an overlap region is only created in areas with a minimum eigenvalue above a predetermined threshold.

1.1.1 Deciding distribution and number of point-pairs

After subdividing the mesh, we must determine how to sample the corresponding point-pairs in the overlap region for use in global registration, to ensure that the compensating warp we introduce is correct.

There are several options for choosing a distribution of sample point-pairs. The first is to use a uniform distribution. However, Rusinkiewicz showed that a major disadvantage of using uniform distribution is that some features may be undersampled, leading to an inability to correctly determine components of the relative rigid-body transformation [11]. Thus, a better option is to use normal-space sampling, also proposed in [11]. However, normal-space sampling may not sufficiently minimize rotational uncertainty in registration, so an even better option is to use covariance-based sampling [6]. In the present paper, we use this covariance-based approach.

The number of corresponding point-pairs chosen is also important. As we said in the beginning of Section 1.2.2 on sizing the overlap, the overlap between two meshes can be considered a hinge. We would like to keep the compensating warp controlled; thus, we would like to create stiff hinges between regions originating from the same mesh. The global registration method we use is a least-squares method. Thus, a higher sampling of the overlapping region between two meshes will constrain their motion to a higher degree, creating a stiffer hinge. Hence, we increase the number of corresponding point-pairs we sample for global registration between overlapping regions originating from the same mesh to a number larger than the number used between other overlapping meshes; in practice, we typically double this number.

1.1.1 Eliminating biases due to overscanning

In areas of the object that were scanned more extensively than other areas, many meshes will overlap that area, and hence more point-pairs from that area will be considered during global registration. We believe this bias is an underappreciated source of error in global registration algorithms. This problem is compounded by our technique, because we effectively create additional meshes, which will increase the number of point-pairs considered during global registration (even if we did not sample double the number of point-pairs between regions originating from the same mesh). The distribution of these new meshes over the area of the object may be uneven, and thus the distribution of point-pairs over the object may also change unevenly.

To avoid this bias, our variant of global registration normalizes the number of point-pairs based on the object. Ideally, we would like to normalize by surface area, but this is difficult to implement. Hence, our approximation normalizes by volume. This is a good approximation if the surfaces do not fold too quickly, which is true for scanned surfaces. Our strategy is to break the axis-aligned bounding box containing the set of rigid meshes into voxels. Each voxel will only accept a set maximum number of point pairs (we typically use 1000 points per voxel). Global registration is then run on this subset of the original corresponding point-pairs. Figure 6 displays results showing the need for this variant of global registration.

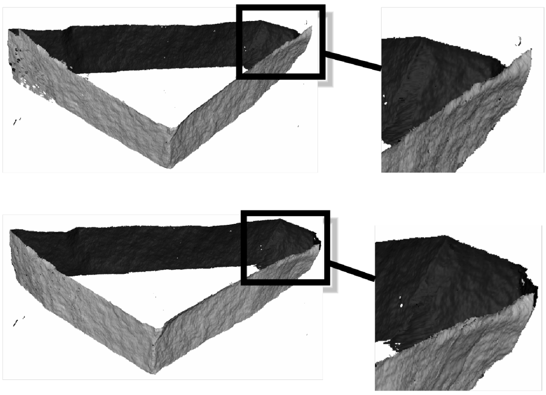

Figure 6. Global registration example showing the need for area-normalization. In the top image, the point-pairs were not normalized. The object has 3-4 meshes overlapping in all but one area (denoted with a square). This area contains only 2 overlapping meshes, and as illustrated in the zoomed image, the meshes have separated. The bottom image illustrates our result with area-normalization.

Chapter 2

Results

"I'd like mornings better if they started later."

- Garfield the cat (by Jim Davis)

In this chapter, we demonstrate our algorithm running on real data. In each case, we will display the mesh resulting from aligning and merging the original (warped) meshes, and the mesh resulting from applying a compensating warp. We evaluate the quality of our alignment by computing the average of the 10% of point-pairs over the object that exhibit the greatest point-plane distance.

2.1 Results on sample Digital Michelangelo project data

Figure 7 shows a set of meshes from the Digital Michelangelo project capturing the face of Michelangelo's David. The top half shows a mesh resulting from aligning and merging the face under incorrect calibration. The inset exhibits a particularly noticeable artifact about the lips. Our error metric computed over the original set of meshes that created this mesh was 1.15 mm. The bottom displays the mesh resulting from adding compensating warp. The inset shows that the objectionable artifacts present in the original mesh have now largely disappeared. Our error metric computed over the diced set of meshes is roughly 0.8 mm. Both meshes were scanned at 0.5 mm resolution.

2.2 Results on sample Forma Urbis Romae project data

Figure 8 shows a set of meshes from the Forma Urbis Romae Project (refer to the project website at http://graphics.stanford.edu/projects/forma-urbis). The Forma Urbis was a marble map of the city of Rome carved during ancient times. During the fall of the empire, the map broke into fragments, some of which still exist today. A set of 4 scans captures the top of fragment 033abc.

Figure 8(a) shows the mesh resulting from aligning and merging the 4 scans under slightly incorrect calibration. Our error metric under incorrect calibration evaluates to 0.787 mm. Figure 8(b) contains the mesh resulting from adding compensating warp. Our error metric now evaluates to 0.395 mm, a 2x reduction in the misalignment. The spacing between range samples in both meshes is 0.5 mm.

The inset in Figure 8(a) exhibits an artifact arising from misalignment error during global registration. The lines are warped and blurred, and appear very different from the photograph displayed. Our algorithm reduced this artifact; the result is shown in the inset of Figure 8(b). The running times for each step of the algorithm executed on an Intel Pentium 4 2.80 GHz processor are displayed in the table in Figure 9. The running times show that the time to dice each mesh at every iteration is small compared to the expensive pairwise matching step.

Thus, in this Forma Urbis case, we have reduced misalignments by a factor of roughly 2x, placing them close to the resolution of our scanner.

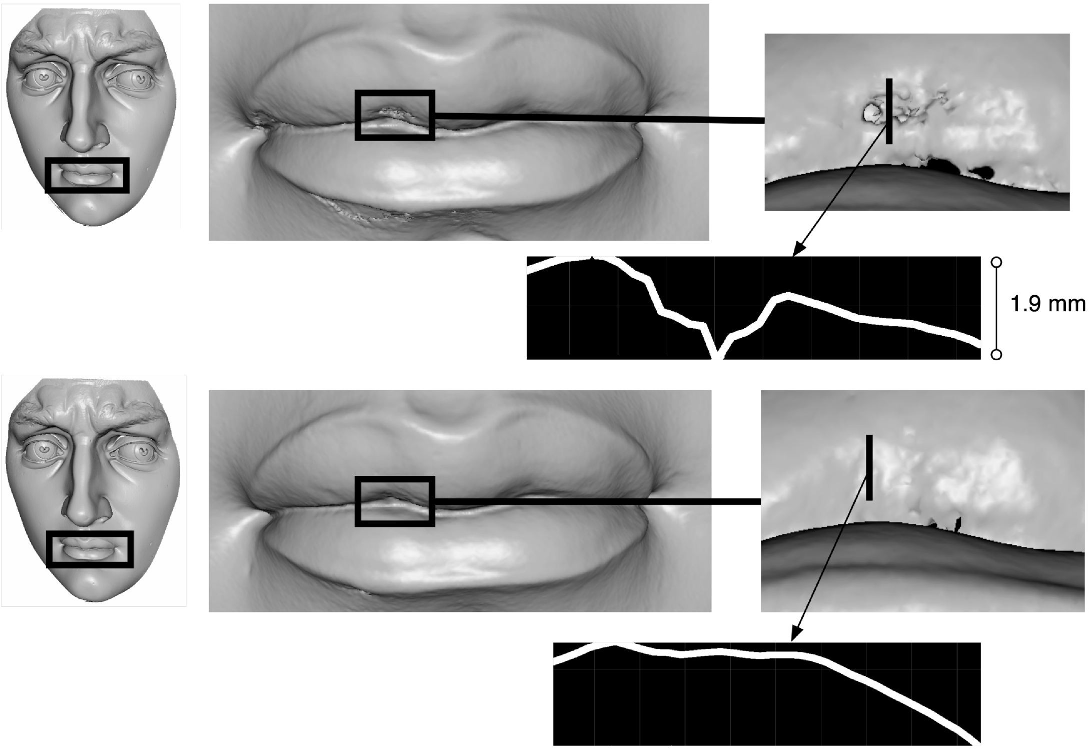

Figure 7. Results on the face of Michelangelo's David. Top: alignment and merging under incorrect calibration. The lips exhibit gross misalignments on the order of 1.9 mm. Bottom: our results. The artifacts in the lips have been greatly reduced.

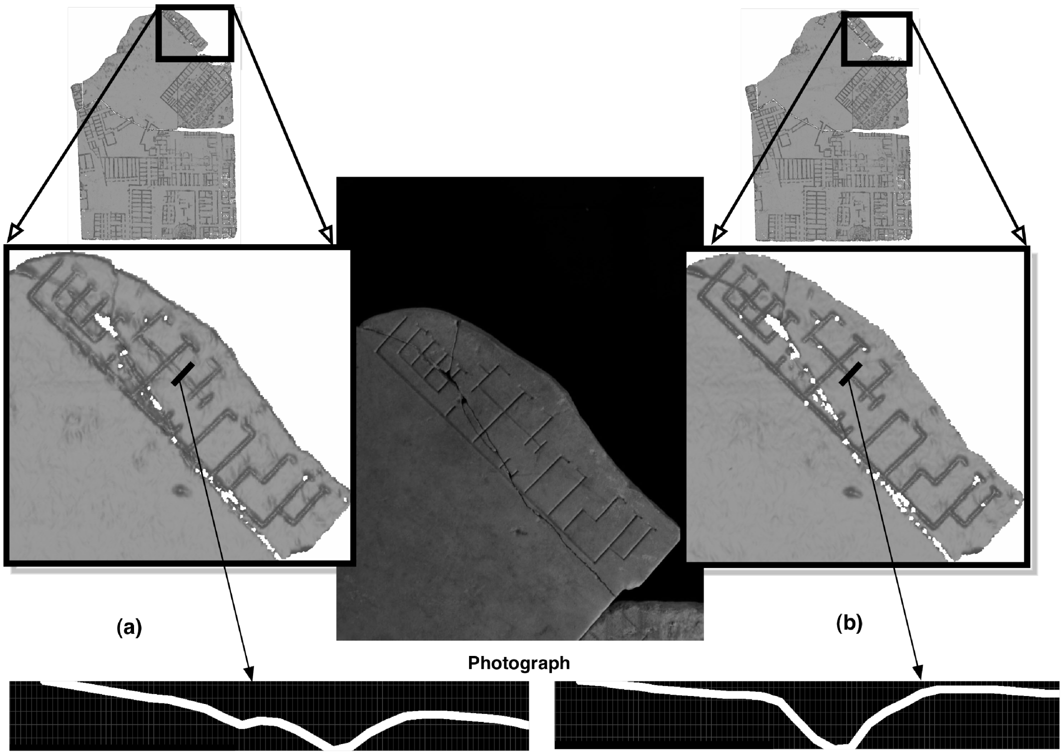

Figure 8. Results on a marble fragment of the Forma Urbis Romae. Left: computer rendering of mesh from alignment and merging under incorrect calibration. Left inset: blurring of the map due to incorrect alignment. Plot shows that a sample incision is shallow (at a depth of 0.25 mm) and incorrectly shows two minima. Right: results of our technique on same input scan. Right inset: the blurring is greatly reduced, and the depth of the incision measures 0.5 mm, which is more consistent with the approximate depth of the Forma Urbis incisions.

Regions	Polygons (million)	Pair Match.	Global Reg.	# points selected	Dicing
4	31	3:09	0:02	1200	0:22
37	31	10:33	0:02	25,800	2:15
75	35.8	34:00	0:04	26,200	2:30
117	40.2	21:00	0:05	63,510	3:30
160	46.1	20:00	0:06	123,110	3:00
197	52.2	17:00	0:13	204,268	

Figure 9. Breakdown of running times by step for Forma Urbis case. We begin with 4 meshes, which we break into 37 scanner sweeps. Times are given in minutes:seconds. The minimum size of the resulting meshes was 15 cm along the longest dimension. The size of the original fragment is approximately 85 cm by 120 cm.

2.3 Discussion of results on sample Forma Urbis Romae project data

One of the motivating goals behind this work is to automatically align the scans in the Forma Urbis Romae dataset to a high degree of precision, in order to produce high-resolution 3D models of all of the fragments. We wish to add these high-resolution 3D models to a fragment database already containing historical information and color data of each of the fragments (refer to http://formaurbis.stanford.edu/docs/FURdb.html); this database will eventually be made publicly available and serve as a resource for scholars.

The majority of the fragments scanned with our custom-built scanner (used on the Digital Michelangelo project and modified for the Forma Urbis Romae project) remain poorly aligned, with errors typically on the order of a few of millimeters. Misalignments usually manifest themselves as false geometry in the final merged mesh. Unfortunately, false geometry can be difficult to distinguish from the true geometry representing the actual shape of the object, and thus may be confusing or misleading to scholars using the 3D models to study the fragments. Thus, it is clear that the alignment of the scans must be improved before they can be merged into final models and included in the database. However, there are 478 unaligned fragments, making it prohibitively time-consuming to improve the alignment of all the fragments manually. Thus, it was hoped that an automatic procedure to align the scans could be formulated.

The method discussed in this paper has yielded a technique for automatically aligning the scans capturing the top surfaces of these fragments. We have demonstrated our technique on the top surface of one of the larger fragments in Section 2.2. Having sufficiently accurate models of the top surfaces are of particular importance to scholars because the map data itself was carved into the top surface.

However, the method discussed in this paper alone is insufficient to align the complete models of the fragments. This is due to the fact that there are other problems in the scans besides warp from miscalibration.

One of the major problems is grazing-angle data. Ideally, the scanner should always be oriented perpendicularly to the surface being scanned. However, in some scans, parts of the surface may be captured with the laser making an acute angle with the surface. There are two complications that arise in these cases. First, all laser triangulation scanners record poor data at oblique scanning angles because the laser beam spreads out on the surface (refer to the appendix of [10]); second, subsurface scattering in the marble compounds this problem by introducing bias and noise into the process of sensing the spread-out beam [7]. Figure 10 demonstrates our measurement strategy for the Forma Urbis fragments. Most of the vertices in the scans capturing the upper and lower edges were scanned at an oblique angle, making two out of the five scans for every fragment problematic to align because the point samples and normals (which are derived from the points) are unreliable.

There are two classes of techniques we could use to address this issue. We can either try to fix the unreliable data, or we can discard it.

If we wish to try to fix the data, we could blur each vertex based on scan angle. Before alignment, we could convolve each vertex with a blurring filter whose radius is directly proportional to the scan angle. Since the normals are derived from the point-samples, this would have the effect of blurring both the geometry and the normals. This is an area of future work.

If we wish to discard the unreliable data, we can discount, or alternatively downweight, vertices captured at a grazing angle. We have developed and experimented with some such techniques.

Specifically, we have implemented a scheme for discounting vertices, in which vertices scanned with an angle greater than a predetermined threshold are discarded. The scan is then retriangulated. However, it is unclear how the threshold should be chosen since a low threshold may discard too much data, and a high threshold may allow too much false data. In practice, as illustrated in Figure 10, we discard vertices where the dot product between the normal and the scan direction is less than 0.65, which roughly corresponds to an angle of 50 degrees. Another consideration is that since data is discarded, holes may be present in the final mesh. These holes can either be ignored or plausible geometry can be interpolated with a hole-filling algorithm.

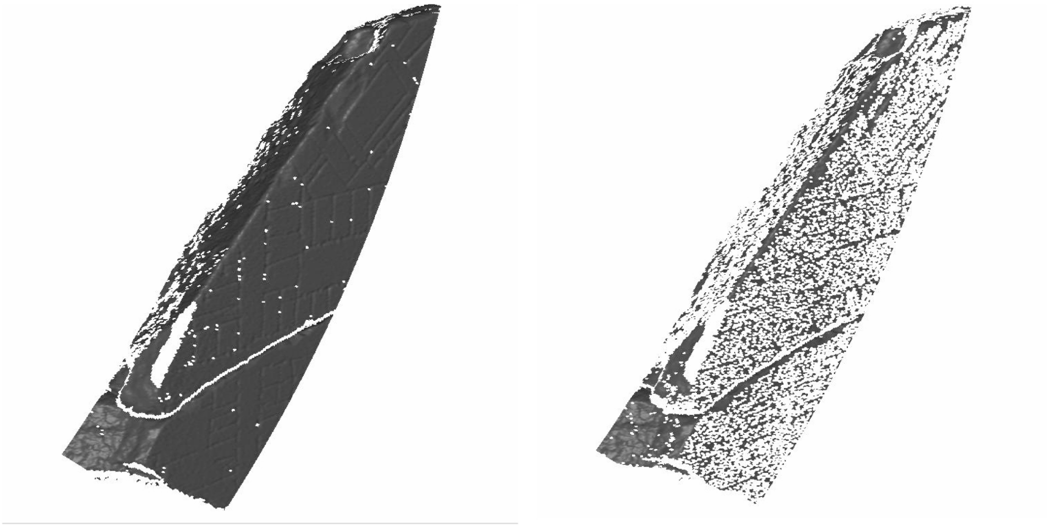

Figure 10. Sample scan demonstrating our filtering technique. The scan on the right is the filtered version of the scan on the left, where we filtered all vertices where the dot product between the normal and the scan direction was less than 0.65, which roughly corresponds to an angle of 50 degrees. As this figure demonstrates, much of the scan data is filtered out, meaning many of the vertices were scanned at an acute angle to the surface and were thus unreliable samples.

Still another technique we considered is to downweight the unreliable data during merging. Because we consistently used one particular measurement strategy (shown in Figure 11), we know that the scans we labeled *up* and *down* exhibit unreliable points and normals. Thus, after alignment, we can downweight these scans during interpolation to have a very low weight (the exact level of down-weighting does not matter), so that the algorithm only uses data from these scans when it has no other choice. The difficulty with this approach is that if misalignments are present, bulges can form where the *top* scan ends and no longer overlaps the *up* scan (and analogously where the *bottom* scan no longer overlaps the *down* scan). The bulges form because the scans are no longer averaged when they overlap; instead, because of this weighting scheme, there is a discontinuity (or large "step") in the interpolation.

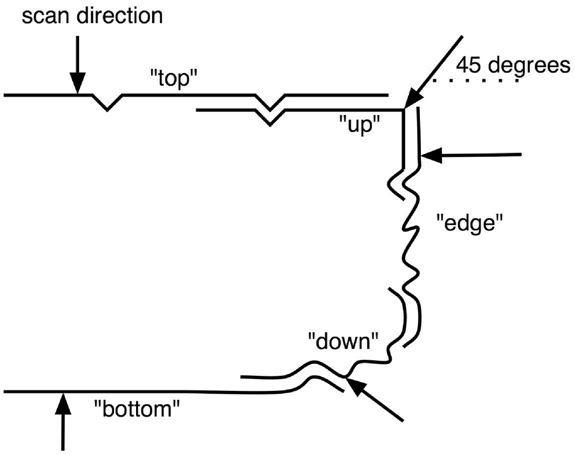

Figure 11. Diagram showing our measurement strategy for the Forma Urbis Romae fragments. The labeled lines indicate cross-sections of the scans that were taken. The *top*, *bottom*, and *edge* scans were captured with the scanner oriented perpendicularly to the surface. The corners were captured with the scanner angled at 45 degrees away from the normal to the top surface for the *up* scans, and 45 degrees away from the normal to the bottom surface for the *down* scans. Because the *up* and *down* scans were captured at an oblique angle, the points and normals (which are derived from the points) are generally unreliable in these scans, making it problematic to align them correctly to the rest of the scan set.

A final alternative is to simply discard the *up* and *down* scans after alignment and before interpolating the final mesh. For archaeological research on the Forma Urbis, accuracy on the top surfaces is very important. Misalignments between the *up* and *top* scans typically cause false geometry (such as double incisions) in the final mesh. Thus, while we would like to make our models as complete as possible, our first priority is to make the top surfaces accurate. Hence, sometimes we are forced to trade off completeness of our models for accuracy of the top surface by simply deleting the *up* and *down* scans.

Figure 12 demonstrates each of these techniques on the scan set capturing the shape of fragment #037Aac, and notes the common artifacts produced with each technique. We have found that by using these techniques to align grazing-angle scans, coupled with mesh dicing for the larger fragments, we have been able to align and merge models for the Forma Urbis database.

However, for many fragments, the underlying problem of misalignment between the top and edge surfaces remains regardless of the alternative chosen. In these fragments, there seems to be not enough data to form the correct relationship between these two surfaces. In the next chapter on *Conclusions & Future Work*, I will make some recommendations specifically for the Forma Urbis project about this issue.

In addition to the techniques discussed above, we could try to downweight vertices gradually with scan angle; this down-weighting scheme could be implemented in ICP or during merging. In ICP, we could change the sample rate smoothly with scan angle. Alternatively, we could weight each vertex proportional to scan angle during merging by setting a confidence value. We attempted this solution, but there are two implementation blocks. First, VRIP works off an orthographic assumption, so the confidence values for the vertices must be set prior to merging. Second, setting the confidence prior to merging conflicts with our current implementation of accounting for edge curl. Specifically, VRIP normally will discount vertices within a certain number of edges away from the boundary since these vertices exhibit bias. However, if the confidence of all vertices is set prior to VRIP, this functionality is disabled. Hence, the implementation of these schemes is another area of future work.

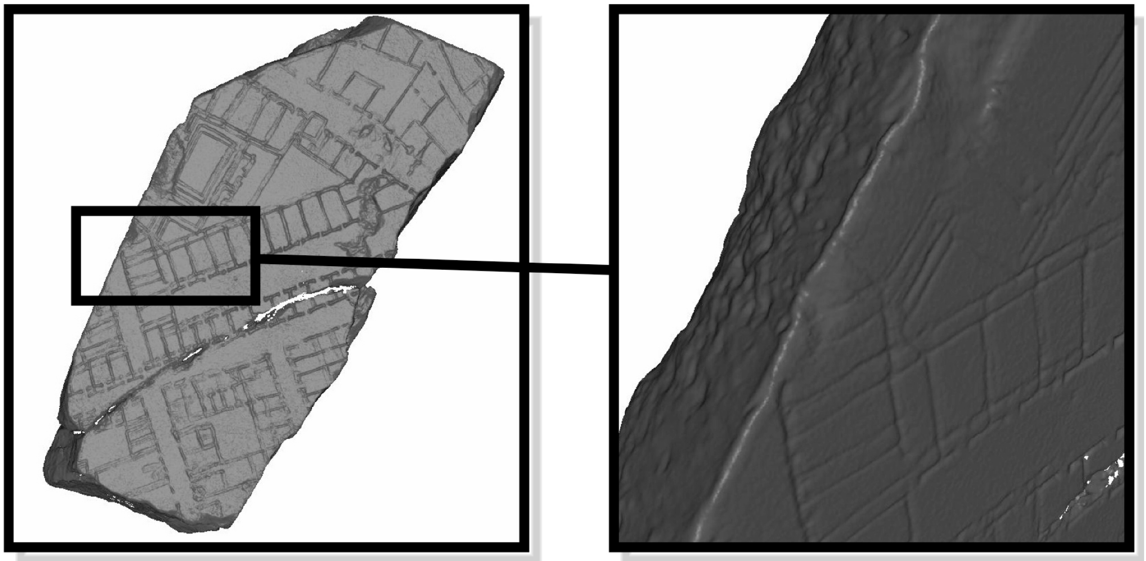

Figure 12(a). Mesh of Forma Urbis fragment #037Aac exhibiting misalignment artifacts from grazing-angle scans. (This fragment is approximately 80 cm x 40 cm on the top, which is smaller than the size at which we typically witness warp from miscalibration of our particular scanner.) Note the incorrect geometry on the top surface, manifested by double-incisions.

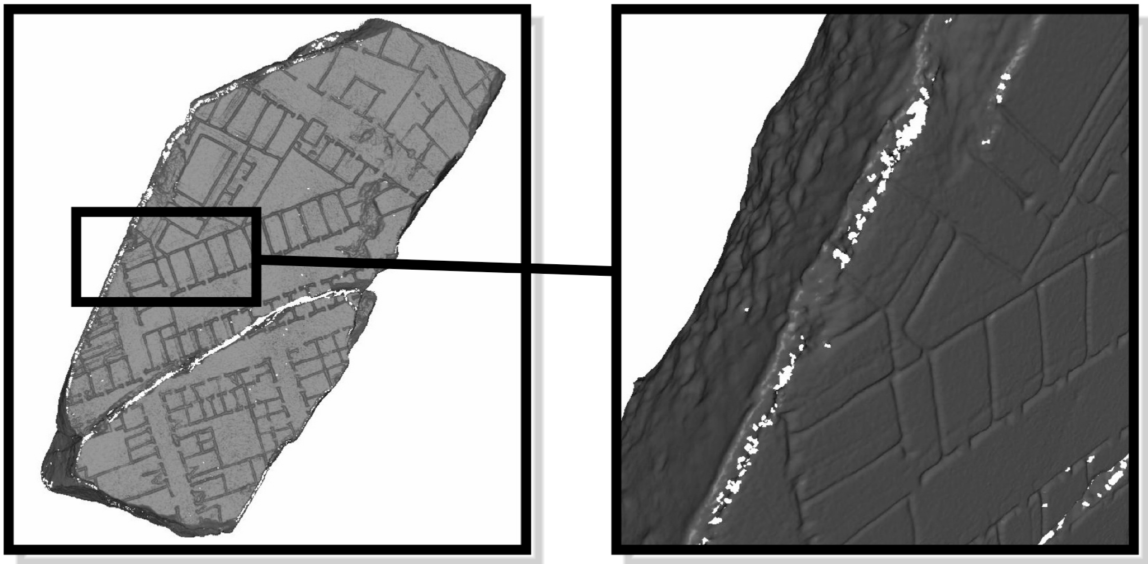

Figure 12(b). Results from filtering the scans by scan-angle. Note that there are still some faint double incisions present. More aggressive filtering would eliminate these artifacts, but there would also be too little data to align the *top* scan to the *up* scan correctly.

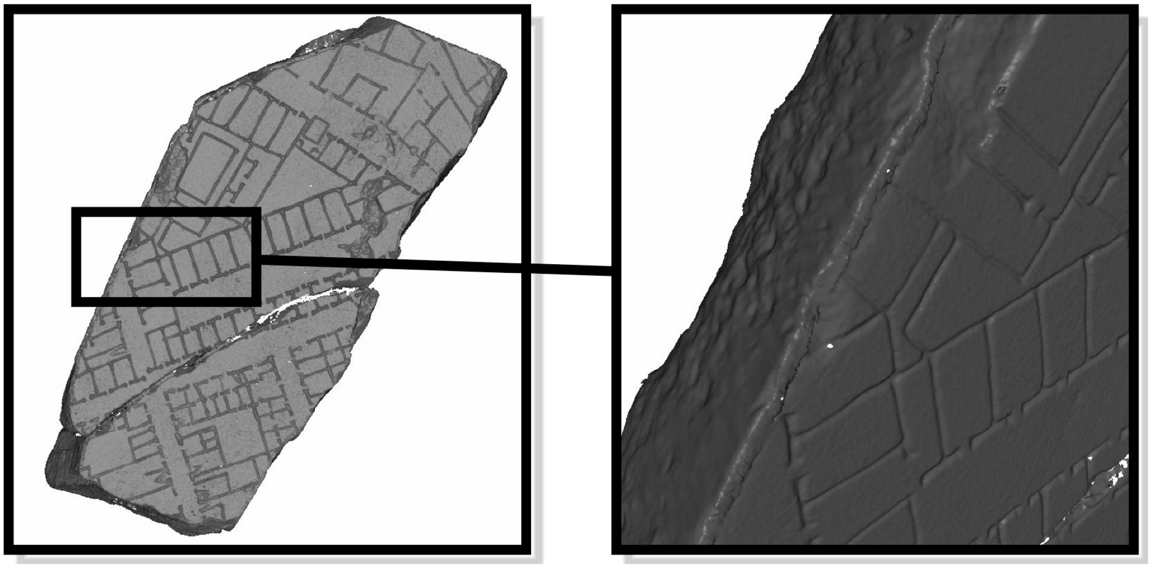

Figure 12(c). Results from downweighting the *up* and *down* scans during interpolation. The double incisions are no longer present, but a 0.7 mm misalignment between the *top* scan and the *up* scans has caused a bulge on parts of the rim of the fragment. (A similar bulge exists on parts of the bottom rim of the fragment.)

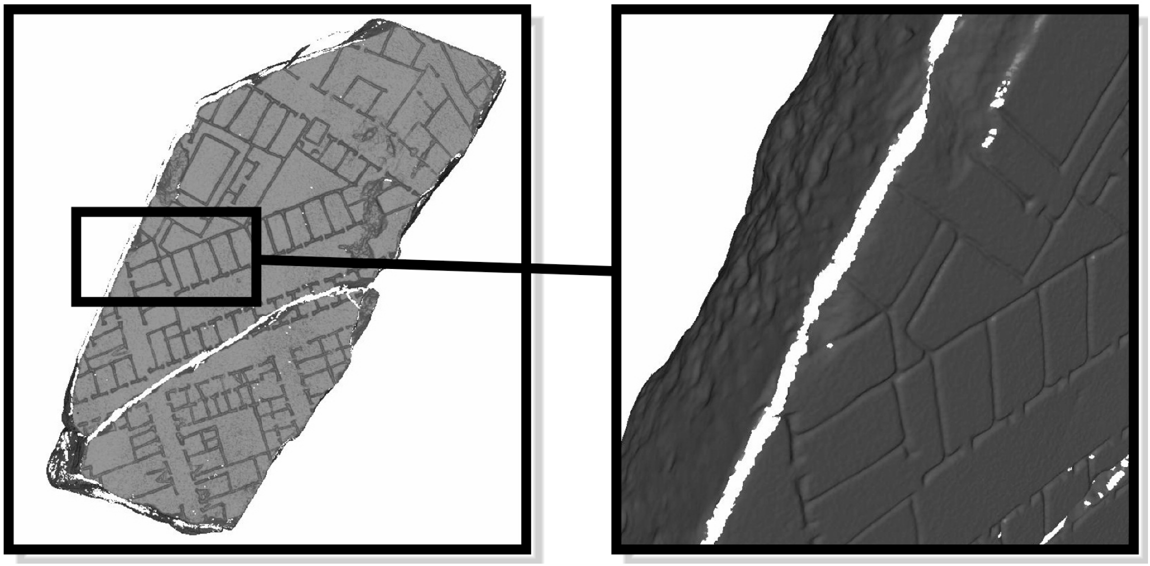

Figure 12(d). Results from deleting the *up* and *down* scans before interpolation. The double incisions are no longer present, and there is no false geometry introduced, but there are significant holes in the final mesh.

Chapter 3

Conclusions & Future Work

"Don't worry about the world coming to an end today. It's already tomorrow in Australia."

- Charles Schulz

In this paper, we presented a method for making mesh alignment techniques more robust to scanner miscalibration. Previous methods that computed rigid transformations only are robust to noise, but may fail or produce incorrect alignments in the face of warp. We presented a method for introducing a minimal compensating warp in the mesh data to allow convergence. This method does not require a specific characterization of scanner warp, and is relatively simple to implement.

One limitation of our proposed method pertains to our choice of compensating warp. We choose a piecewise rigid model that in the limit converges to a curved warp. However, if scanner miscalibration introduced a warp that is not curved, this method will not converge.

A second limitation of our method is that we do not guarantee that the shape of the object converges to the correct answer. We introduce a controlled, minimal amount of warp and ensure geometric constraints are met so that the resulting shape will not diverge. However, without independent observations to constrain the shape of the object, it is difficult to compare the shape resulting from our method with the actual shape of the original object. In the Digital Michelangelo project, we obtained independent measurements of several points on most of the statues scanned using a theodolite (a surveying tool that locates points in 3-space). As future work, we can incorporate this data as additional constraints on the shape of the object [1].

A third set of limitations pertains to the design of the algorithm. Currently when creating sub-meshes, we set the overlap to be a fixed percentage of the oriented bounding-box. However, this overlap may be of an incorrect size: it may be too large and thus overconstrain the motion of its constituent meshes during global registration, or it may be too small and may not constrain the motion of its constituent meshes enough. Hence, it seems as though we should grow the overlap region until it contains enough features to render the matching stable, e.g. using the machinery to analyze stability that appears in our accompanying paper [6].

One intriguing area of future work lies in global registration itself. It seems advantageous to redistribute the sampling of corresponding point-pairs in this step based on a one mesh to many meshes stability analysis. Currently the point-pair distribution is decided in ICP considering only a one-mesh to one-mesh stability relationship. However, during global registration we could consider the stability of a mesh in relation to all of its overlapping partners, as shown in Figure 13.

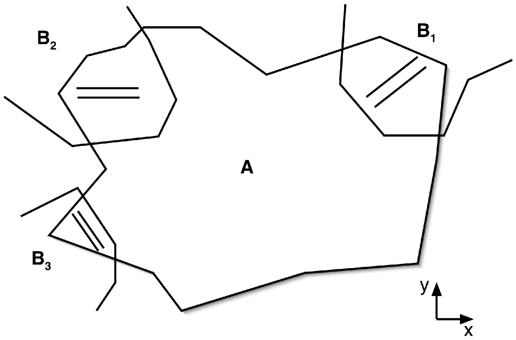

Figure 13. A diagrammatic example demonstrating the need for re-distributing the point-pairs in global registration based on a one-to-many stability analysis. The parallel lines indicate features, e.g. incisions in a Forma Urbis fragment. The features in any one of the B's are insufficient to constrain A's motion, but any two are sufficient. For example, in B_1 the incisions are parallel to the x-axis, so the two meshes are constrained in the y-direction, but are free to slide in x. Similarly for B_2 the meshes are constrained in a single direction, but since this direction is different from B_1, the two will form a 2D basis, constraining sliding in the plane.

Also, as mentioned in the introduction, our method implicitly converges in the limit to a curved warp. However, we could instead explicitly fit a warp function to our data. We could first fit a spline to each of the meshes. The spline data would show us the directions in which the meshes are being warped. Then if we subsequently ran the technique discussed in this paper, we could modify the distribution of points to allow movement to compensate for these known warps.

Finally, an improved measurement strategy can reduce the effect of warp and generally increase the accuracy of alignment. As noted in Section 1.2.2, a set of meshes can be viewed as a mechanical system. By analyzing this system, we conjecture that a scheme for view-planning can be devised. Also, knowledge about the scanner itself, such as which axes are less accurate than others, can help improve the measurement strategy.

Recommendations for the Forma Urbis project data

For the Forma Urbis project, it may not be possible to align the top surface to the edge surface (and the bottom surface to the edge surface) for many of the fragments if the data in the *up* and *down* scans is too unreliable. If techniques such as filtering the *up* and *down* scans (which has been implemented) and blurring (which has not) do not fix the alignment, two alternatives discussed in the previous section are to downweight the *up* and *down* scans during interpolation or to delete the *up* and *down* scans before interpolation.

In some cases, we may wish to produce two models of the fragment. We could produce a lower-resolution model containing all of the scans, with the error distributed evenly. This model could be used for matching algorithms that require judging the thickness of the fragment or the angle of the top surface to the edge surface. Then, since accuracy of the top surface is very important for this project, a high-resolution model could be produced where the top surface is as error-free as possible. This second model could be used for matching algorithms that require very accurate data on the incisions.

References

[1] J-A Beraldin, L. Cournoyer, M. Rioux, F. Blais, S.F. El-Hakim, and G. Godin, "Object model creation from multiple range images: acquisition, calibration, model building and verification," *Proc. 1st Int'l Conf. on 3-D Digital Imaging and Modeling,* IEEE, 1997, pp. 326-333.

[2] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a General Multi-view Registration Technique", *IEEE Trans. Patt. Anal. Machine Intell.,* 18(5): pages 540-547, May 1996.

[3] P.J. Besl and N.D. McKay, "A Method for Registration of 3-d Shapes", *IEEE Trans. Patt. Anal. Machine Intell.,* 14(2): pp. 239-256, 1992.

[4] Y. Chen and G. Medioni, "Object Modelling by Registration of Multiple Range Images", *Image and Vision Computing,* 10(3):145-155, 1992.

[5] A. Fusiello, U. Catellani, L. Ronchetti, and V. Murino, "Model Acquisition by Registration of Multiple Acoustic Range Views", *Proc. EECV*, pp. 805-822, 2002.

[6] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically Stable Sampling for the ICP Algorithm", to appear in *3DIM 2003.*

[7] G. Godin, J.-A. Beraldin, M. Rioux, M. Levoy, L. Cournoyer, F. Blais, "An Assessment of Laser Range Measurement of Marble Surfaces", *Proc. Fifth Conference on optical 3-D measurement techniques*, Vienna University of Technology, Vienna, Austria, 2001.

[8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions", *Journal of the Optical Society of America A,* 4(4): 629-642, 1987.

[9] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, "The Digital Michelangelo Project: 3D scanning of large statues", *Proc. SIGGRAPH 2000,* ACM SIGGRAPH, pp. 131-144.

[10] K. Pulli, "Multiview Registration for Large Data Sets", *Int'l Conf. on 3D Digital Imaging and Modeling,* pp. 160-168, 1999.

[11] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm", *Proc. 3DIM,* pages 145-152, 2001.

[12] H-Y Shum and R. Szeliski, "Panoramic Image Mosaics", *Microsoft Research*, Technical Report MSR-TR-97-23.

[13] A.J. Stoddart & A. Hilton, "Registration of Multiple Point Sets", *Proc. 13th IAPR Int'l Conf. on Pattern Recognition,* pp. 40-44, 1996.

[14] R. Szeliski and S. Lavallee, "Matching 3-D Anatomical Surfaces with Non-Rigid Deformations using Octree-Splines", *IEEE Workshop on Biomedical Image Analysis,* pp. 144-153, 1994.