A Hierarchical Method for Aligning Warped Meshes

Leslie Ikemoto
Stanford University
leslie@cs.stanford.edu

Natasha Gelfand
Stanford University
ngelfand@cs.stanford.edu

Marc Levoy
Stanford University
levoy@cs.stanford.edu

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 paper, 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.

1. Introduction

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 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], [9], and [12].

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. Moreover, it can be very difficult to design a 3D scanner that can accurately capture high detail (the Digital Michelangelo project required a resolution of 0.25 mm) on a large object (such as the David at 5 m tall) [8]. 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.

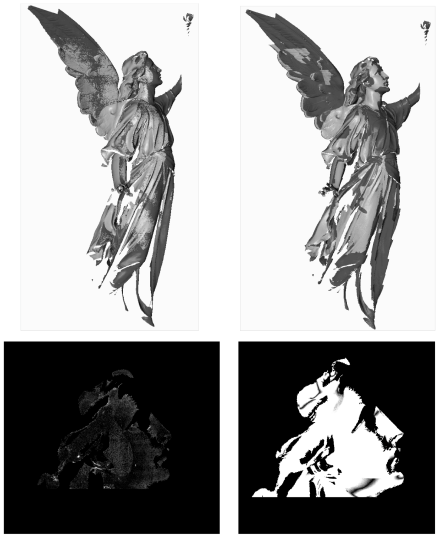

Figure 1. An example of two meshes that exhibit warp. Top left: Two scans acquired under correct calibration; average point-to-plane error over 200 corresponding point-pairs is 0.13 mm. Bottom left: a depth disparity map of the face of the angel (black indicates 0 mm depth disparity between the meshes and white 1.0 mm disparity). Top right: the same 2 scans aligned under simulated incorrect calibration; the average error 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 [8], we often saw this much misalignment due to miscalibration of our scanner.

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.

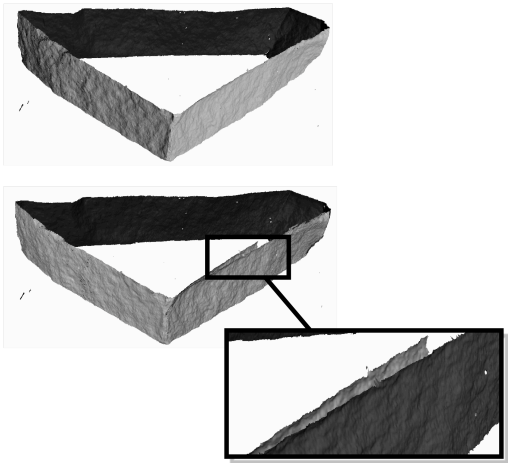

Figure 2. A multi-mesh set exhibiting warp. Top: the scans were acquired under correct calibration, and properly form a closed loop. Bottom: the scans are misaligned due to warp from scanner miscalibration. Inset: 4 mm misalignment between 2 scans (colored different shades of gray). Misalignments during 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.

One of the motivating goals behind this work is to automatically align the scans in the Forma Urbis Romae dataset (refer to the project website at http://graphics.stanford.edu/projects/forma-urbis) to a high degree of precision. The Forma Urbis was a marble map carved during ancient times of the city of Rome, measuring 60 feet wide and 45 feet high. During the fall of the empire, the map broke into fragments, some of which still exists today. The problem of piecing these fragments together is still unsolved, and made difficult by the large number and heavy size of the fragments. Archaeologists could instead use 3D models of the fragments to help put the map back together, but in order to be of use these models must have a high level of accuracy at a high resolution, especially on the top surfaces where the map data is incised.

Of the 1,186 fragments we scanned, 481 were acquired using the custom-built laser triangulation scanner used on the Digital Michelangelo project and modified for the Forma Urbis Romae project. Unfortunately, due to miscalibration of this scanner during the acquisition process, most of these fragments remain poorly aligned, with errors typically on the order of a few of millimeters. Misalignments usually manifest themselves as false geometry (e.g. double incisions) 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. However, 477 fragments have not yet been successfully aligned, 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.

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. 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.

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 [11]. 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 [13]. We discuss using this technique as an extension to our method in Section 5. 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 2 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 3.

2. 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.   [P1, P2] = dice(P_worst)
j.   if (!diced_P_worst_successfully) break
k.   [Q1, Q2] = dice(Q_worst)
l.   if (!diced_Q_worst_successfully) break
m.   S.add(P1, P2, Q1, Q2)
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]. There are several variants of the ICP algorithm (refer to Rusinkiewicz's study of these variants [10]). In the Digital Michelangelo project (described in Levoy, *et al.* [8]), we used a variant of ICP devised by Chen & Medioni [4], enhanced by an improved sampling strategy. Instead of uniformly sampling points on P and Q, we use a non-uniform sampling strategy that samples at a higher rate in areas that exhibit more features, and less in areas that appear planar. We choose points based on the covariance matrix between P and Q. This algorithm is described in our companion paper [6].

In step (d), all 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 [2, 5, 9, 12]. In the Digital Michelangelo project, we used Pulli's algorithm [9]. The 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. In this paper, we use a small variant on his algorithm. Specifically, we employ Chen-Medioni's point-to-plane alignment error metric, instead of the point-to-point metric used by Pulli.

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

Finally in (g)-(o), we attempt to 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. Through experimentation, we have found that setting num_pairs_to_dice to a value less than or equal to one-quarter the number of original meshes keeps the compensating warp sufficiently controlled.

In (i) and (k), we dice the meshes we selected. Section 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 3. Steps (j) and (l) check whether the meshes were subdivided successfully. We then 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.

Figure 4 demonstrates the amount of compensating warp introduced by our technique on a sample data set from the Forma Urbis Romae project.

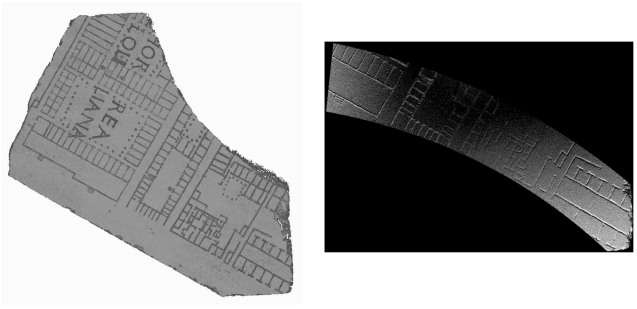

Figure 4. Visualization of the amount of warp introduced in a sample data set (i.e. the top surface of Forma Urbis Romae fragment #025a). In order to produce the mesh on the left, the 13 scans capturing the top surface of the fragment were diced into 84 sub-meshes and hierarchically aligned. The visualization on the right-hand side shows a depth disparity map between a representative scan and the 7 sub-meshes this scan was diced into after hierarchical alignment. (Black indicates 0 mm depth disparity between meshes and white 1.0 mm disparity.) The scan has been warped more toward the edges than the middle.

3. 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.

3.1 Subdividing a mesh

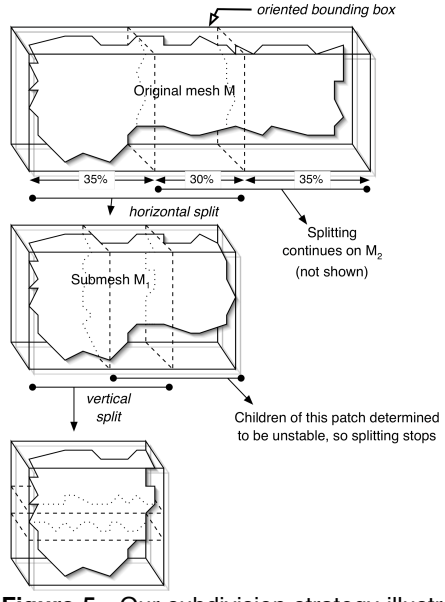

Figure 5. Our subdivision strategy illustrated on a sample mesh M. M is first subdivided along the horizontal direction into M_1 and M_2. Submesh M_1 is then later split horizontally again. The right child is determined to be unstable, so splitting stops. The left child is later split vertically.

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 5. 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.

Note that Figure 5 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. 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, since we conjecture it will outperform 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, 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 anywhere on the mesh.

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].

3.2 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 standpoint, we can view the overlap between two meshes as a hinge, where the stiffness is proportional to 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 6 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 30% the size of the original meshes (i.e., we set the volume of the overlapping region's oriented bounding box to 30% the volume of B).

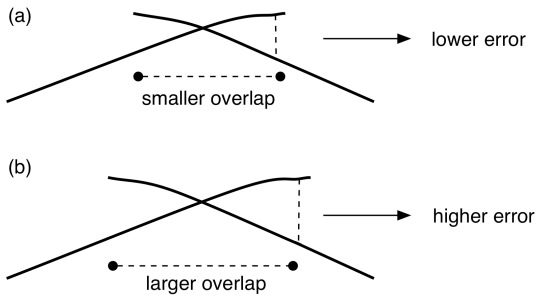

Figure 6. Example showing that overlap size determines the "stiffness" of the hinge between 2 overlapping meshes during global registration. The meshes in (a) and (b) have the same relative rotation, but since the overlap in (b) is larger than in (a), the point-to-plane error in (b) is higher. Thus, the hinge in (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- or 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 use a stability analysis, analyzing the eigenvalues of the covariance matrix between two meshes [6]; an overlap region is only created in areas with a minimum eigenvalue above a predetermined threshold, in the hope that such an area will have at least one lock-and-key feature.

3.3 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.

Two important considerations are the spatial distribution of point-pairs, and the total number of pairs. There are several options for choosing a distribution of sample point-pairs. We minimize uncertainty in translation and rotation during registration by using covariance-based sampling [6].

Regarding the number of point-pairs, we would like to keep the compensating warp controlled. The global registration method we use is a least-squares method, so a higher sampling of the overlapping region between two meshes will constrain their motion to a higher degree. The proper value for the number of corresponding point-pairs chosen for sub-meshes diced from a single mesh should be higher than the number used between sub-meshes originating from different meshes, in order to create a stiffer hinge. In practice, we typically double this number.

3.4 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. 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 since the meshes have not yet been integrated to form a surface. 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 8 displays results showing the need for this variant of global registration.

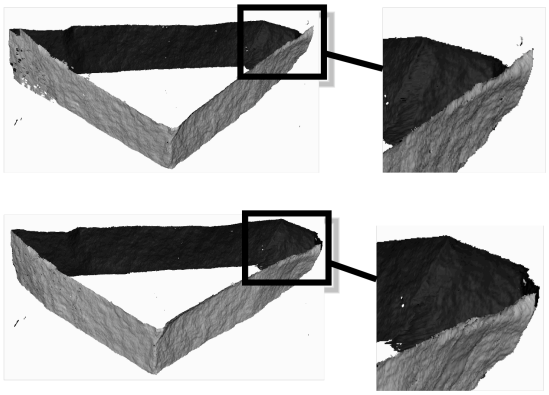

Figure 8. Global registration example showing the need for area-normalization. In the top image, the point-pairs were not normalized. The object has 3-4 meshes overlapping in all but one area (denoted with a square). This area contains only 2 overlapping meshes, and as illustrated in the zoomed image, the meshes have separated. The bottom image illustrates our result with area-normalization.

4. Results

Here we will 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.

Figure 10 shows a set of 4 meshes from the Forma Urbis Romae Project capturing the top of fragment 033abc. Figure 10(a) shows the mesh resulting from aligning and merging the 4 scans under slightly incorrect calibration; our error metric evaluates to 0.787 mm. Figure 10(b) contains the mesh resulting from adding compensating warp by dicing these meshes into 197 sub-meshes. Our error metric now evaluates to 0.395 mm, a 2x reduction in misalignment. The spacing between range samples is 0.5 mm.

The inset in Figure 10(a) exhibits an artifact arising from misalignment error during global registration. The lines are warped and blurred. Our algorithm reduced this artifact; the result is shown in the inset of Figure 10(b). The table in Figure 9 displays the running times for each step of the algorithm executed on an Intel Pentium 4 2.80 GHz processor. The running times show that the time to dice each mesh at every iteration is small compared to the expensive pairwise matching step.

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

Figure 9. Breakdown of running times by step for Forma Urbis fragment #033abc. 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 original fragment is approximately 85 cm by 120 cm.

Figure 11 shows a set of meshes from the Digital Michelangelo project capturing the face of the 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. The spacing between range samples is 0.5 mm.

4. Conclusions

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, but 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 (e.g. In the Digital Michelangelo project, we

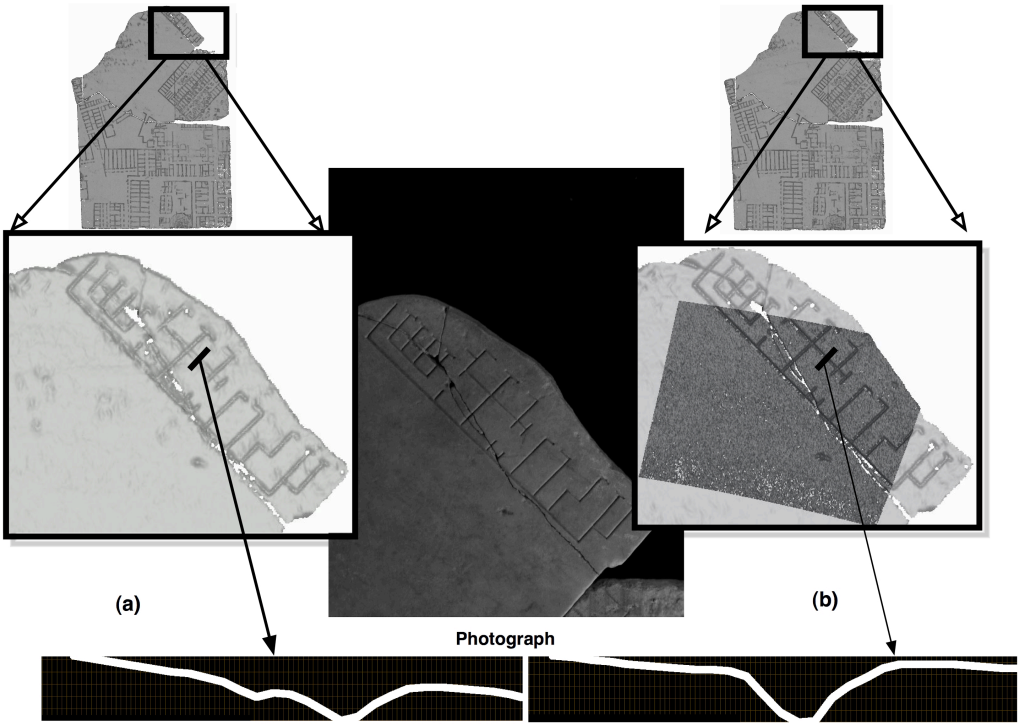

Figure 10. Results on a marble fragment of the Forma Urbis Romae. Left: computer rendering of mesh from alignment and merging under incorrect calibration. Left inset: blurring of the map due to incorrect alignment. Plot shows that a sample incision is shallow (at a depth of 0.25 mm) and incorrectly shows two minima. Right: results of our technique on same input scans. One of the sub-meshes is overlaid in a darker gray. 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.

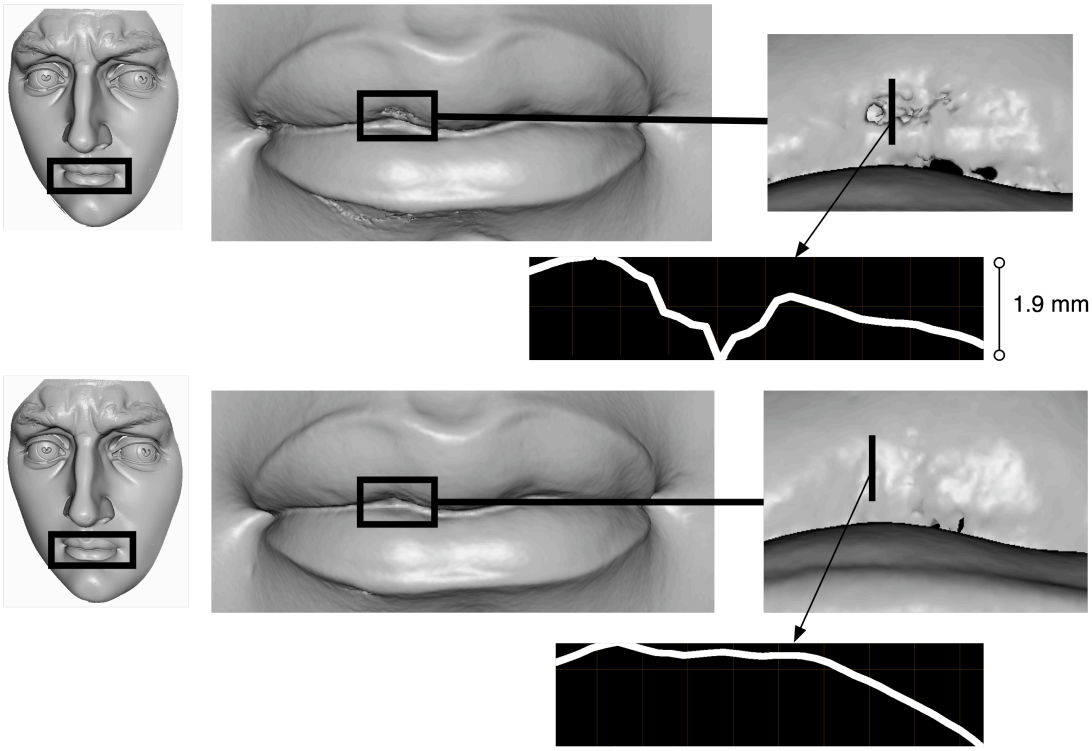

Figure 11. Results on the face of Michelangelo's David. Top: alignment and merging under incorrect calibration. The lips exhibit gross misalignments on the order of 1.9 mm. Bottom: our results. The artifacts in the lips have been greatly reduced.

obtained independent measurements of several points on some of the statues using a theodolite; this data could be incorporated as additional shape constraints [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 overlapping bounding-box. However, this overlap may be of an incorrect size (i.e. 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].

5. Future Work

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 12.

Also, as mentioned in the introduction, our method implicitly converges to a curved warp in the limit as sub-mesh size goes to zero. We could instead explicitly fit a warp function to our data, but as mentioned earlier, we fear that this would run slowly and be unstable. A third alternative, which combines the elegance of a continuous warp function with the speed and stability of our piecewise rigid solution, would be to implement an extension to our technique in which a spline is first fit to each of our 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 3.2, 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. For example, if an object is scanned in vertical strips, such an analysis would probably indicate that another scan should be taken horizontally.

Acknowledgements

We would like to thank our sponsors, National Science Foundation Research Grant IIS-0113427 and the Stanford University President's Fund. Our thanks also go to the rest of the Digital Michelangelo Project and Forma Urbis Romae teams for providing the data on which this work is based.

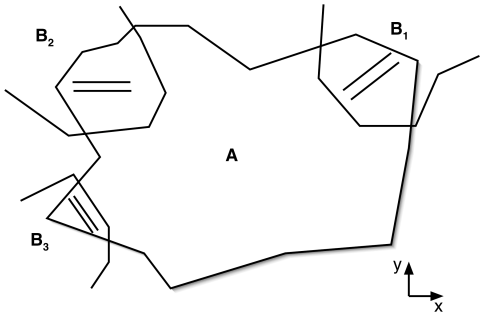

Figure 12. A diagrammatic example demonstrating the need for re-distributing the point-pairs in global registration based on a one-to-many stability analysis. The parallel lines indicate features. The features in any one of the B's are insufficient to constrain A's motion, but any two are sufficient. 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.

References

[1] J-A Beraldin, L. Cournoyer, M. Rioux, F. Blais, S.F. El-Hakim, 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, D. Laurendeau, "Towards a General Multi-view Registration Technique", *IEEE Trans. Patt. Anal. Machine Intell.,* 18(5): pp. 540-547, May 1996.

[3] P.J. Besl & 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 & 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, V. Murino, "Model Acquisition by Registration of Multiple Acoustic Range Views", *Proc. EECV*, pp. 805-822, 2002.

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

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

[8] 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.

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

[10] S. Rusinkiewicz & M. Levoy, "Efficient Variants of the ICP Algorithm", *Proc. 3DIM,* pp. 145-152, 2001.

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

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

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