Finding Surface Correspondences by Local Feature Diffusion

Matthew Fisher

June 11, 2008

Abstract

Establishing corresponding regions between two surfaces is an important part of many problems in surface registration, reconstruction, medical imaging, and object recognition. We present a method that uses diffusion to propagate local feature descriptors across a surface to create a robust signature which can be used for similarity comparisons at varying scales. To collect the local signatures around a point we use an oriented grid which we search using an ϵ-approximate nearest neighbor search algorithm. We use this method to compute a dense set of correspondences between two surfaces that conveys both point and local rigid coordinate frame matches. We present several techniques for aggressively pruning these correspondences to keep only those that are consistent over significant regions.

1 Introduction

Our goal is to compute a mapping between surfaces. The motivating application for this is surface acquisition and modeling. Data is first acquired in the form of a set of point clouds or loosely structured meshes spaced in time, each scan in its own local coordinate frame. Data must then be registered, where we attempt to find a possibly non-rigid transformation that aligns the data in a common coordinate frame. Once the surfaces are aligned in a common coordinate frame we can reconstruct the surface using a standard point cloud reconstruction technique. Other applications include providing an initial alignment for surface flow algorithms [Eckstein 2007],

1.1 Approach

Our approach can use any existing local feature as a base metric. Examples include Gaussian or mean curvature [Gatzke 2005] and integral invariants [Huang 2006]. If the range scans have color information attached to the surface, then we can also use this color information as a base metric for surface matching. We found the best results when using a local feature derived from the behavior of the eigenvalues of the scatter matrix as we enlarge the radius of points used in the fit. Some regions may have very salient features and their local neighborhoods may already match strongly; however, many regions have features that are comparable to the noise in the acquisition process and we cannot expect them to match. We can still define a good local feature value at these points by applying diffusion of progressively increasing scale to the underlying local feature values. Given this improved local feature descriptor defined across the surface, we place an oriented grid on the surface at each point, and in each cell compute the average and standard deviation of this local feature. The values in each grid cell form our final surface signature at a point. We then apply PCA to project this high dimensional descriptor onto a lower dimensional space, and then apply an ϵ-approximate nearest neighbor search algorithm to find good correspondences between surfaces. Because the grids are oriented, each correspondence describes a rigid transformation between the two surface points. Finally, we reject correspondences that are not mutually best candidate matches, and those that do not have a sufficient basis of support.

2 Previous Work

Patch features, SIFT features, spin images.

3 Feature Extraction

The first phase constructs a descriptor for points on each surface. This is done in three principal phases. First, a preprocessing step performs (re-)meshing and samples the resulting mesh surface. Next, we define a local feature value for each vertex on the mesh. To use the language of differential geometry, we will call such a set of values for each vertex a 0-form. We then diffuse this form across the surface, and possibly look at higher order behavior of the feature form with respect to scale by applying the Laplace-Beltrami operator. Finally, we construct a local grid at a fixed orientation for each point, the values of the feature form that fall into each grid cell being used the define the oriented signature at that point.

3.1 Preprocessing

Our algorithm uses a mesh to define geodesic distances and the Laplace-Beltrami operator on the surface. If our input is in the form of a depth map or point cloud, we first construct a mesh representation of the surface using Poisson surface reconstruction. Even if connectivity information to construct a mesh from the point cloud is available, we still found it worthwhile to discard the connectivity information and perform the Poisson reconstruction. This avoids problems where the initial mesh representation may connect regions of the mesh with very poor normal interpolation, and the reconstruction often provides a surface whose normals more faithfully model the underlying surface. Poisson reconstruction performs aggressive hole-filling, generating large regions of surface that are distant from any sample points. Rather than attempting to define a required support distance to existing sample points or otherwise remove these spurious regions, we choose to retain the entire surface as we found the algorithm will often be able to match regions that are relatively distant from any support points, especially if the holes are due to occlusions in the scans, and have very few positive correspondences in spuriously reconstructed regions. The advantage of performing the reconstruction can be seen in figure 1. The algorithm will also make use of uniformly sampled points on the surface. To resample the mesh, we first paramaterize the mesh into a texture atlas and then uniformly sample from triangles in this atlas, taking into account the area distortion to ensure uniform sampling. This produces a very good periodic sampling, without being significantly biased by the underlying triangulation. We extract these samplings at two resolutions. The higher resolution, $\mathbf{S_a}$, is used to construct the local features, and the lower resolution, $\mathbf{S_b}$, is used as seeds for computing correspondences.

3.2 Local Features

The local feature descriptor we use should be invariant to rigid body motion, robust to noise in the surface, and compact in support over the surface. Several such local descriptors are possible. Let r_{local} be the radius of support of the local features. To compute the local descriptor at a surface point $\mathbf{p}$, we first collect all points in $\mathbf{S_a}$ within r_{local} of $\mathbf{p}$. Many techniques have been proposed at this point to construct local features that capture a notion of this local distribution and are robust to sampling density. A popular choice is to look at the eigenvalues of the covariance matrix when we perform PCA on the set of neighboring points. If we denote these eigenvalues $\lambda_1 \leq \lambda_2 \leq \lambda_3$, our local feature descriptor, related to the local surface curvature, is given as

$$f(\mathbf{p}, r_{\text{local}}) = \sqrt{\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}} \tag{1}$$

We can obtain more descriptive features by looking at the first two wavelet coefficents of this function with respect to radius:

$$f_0(\mathbf{p}, r_{\text{local}}) = \int_0^{r_{\text{local}}} f(\mathbf{p}, r)\, dr \tag{2a}$$

$$f_1(\mathbf{p}, r_{\text{local}}) = \int_{\frac{r_{\text{local}}}{2}}^{r_{\text{local}}} f(\mathbf{p}, r)\, dr - \int_0^{\frac{r_{\text{local}}}{2}} f(\mathbf{p}, r)\, dr \tag{2b}$$

We compute these feature descriptors at all mesh vertices, giving us feature 0-forms that will be used in future stages. We denote a 0-form by a column vector $\mathbf{g}$, which has an entry for each vertex. An example of these values on a pair of surface scans is given in figure 1.

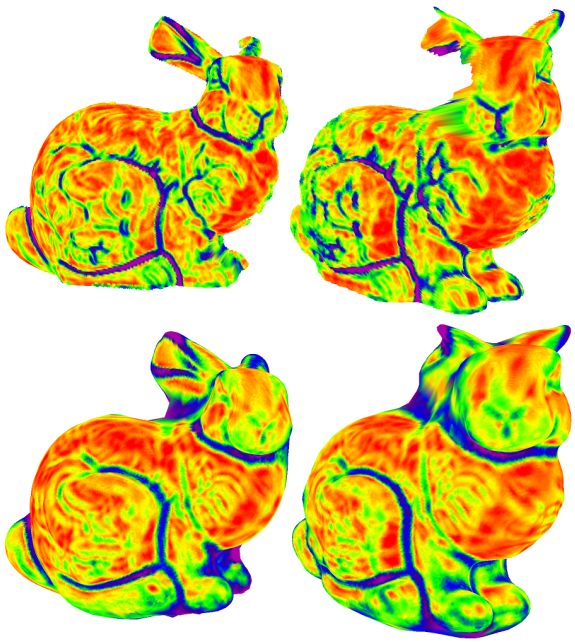

Figure 1: Top: mesh of two surface scans of the Stanford bunny constructed from connectivity information in the point set. Bottom: Poisson reconstruction of the point scans. Both are colored with the value of f_0 at each vertex. The Poisson reconstruction gives a value that is less sensitive to the angle between the angle between the scanner ray and the surface normal.

3.3 Feature Diffusion

Some regions of the mesh will have somewhat unique local descriptors due to the presence of very sharp curvature or other salient features; other regions will have highly uninformative descriptors. Nevertheless these locally undescriptive regions can still be identified by their proximity to nearby significant features. One way to achieve this effect is using a very large value of r_{local}. This is not a good idea as it weakens the descriptive power in regions with significant features. Instead, we perform diffusion on the 0-form from a relatively small r_{local}. For the Laplace-Beltrami operator we use the matrix L derived from discrete exterior calculus although numerous alternative discrete Laplacians could be used in its place. The diffusion process is defined as:

$$\frac{\partial \mathbf{g}}{\partial t} = -\lambda L \mathbf{g} \tag{3}$$

We use implicit Euler's to compute the feature form at future time steps:

$$(I + \lambda \Delta t L)\mathbf{g}_{t+1} = \mathbf{g}_t \tag{4}$$

L can be expressed as $*_0^{-1} d_0^T *_1 d_0$, where $*_1$ and $*_0$ are diagonal matrices and d_0 is the vertex-edge adjacency matrix. This results in an asymmetric system due to the $*_0^{-1}$ term, which is just a diagonal matrix of the triangle areas adjacent to each vertex. The problem can easily be made symmetric by left-multiplying by $*_0$:

$$(*_0 + \lambda \Delta t d_0^T *_1 d_0)\mathbf{g}_{t+\Delta t} = *_0 \mathbf{g}_t \tag{5}$$

Now the matrix in this equation is symmetric positive definite. Furthermore, it depends only on the mesh geometry, which does not change, so we can compute its Cholesky factorization as a preprocess. Thus we can compute a local feature descriptor at any future time by a simple backsubstitution. The behavior of this function with respect to time provides further information about its local neighborhood. Computing the derivatives of this function may be easily accomplished by repeatedly applying the Laplacian:

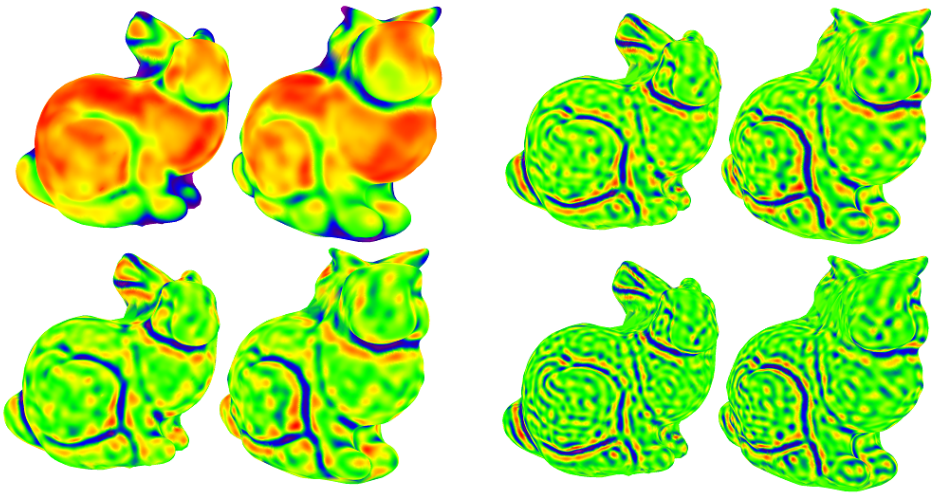

Figure 2: Smoothed version of the 0-form shown in figure 1 (top left), and the 1st (bottom left), 2nd (top right) and 3rd (bottom right) derivatives.

$$\mathbf{g}_t^n = L\mathbf{g}_t^{n-1} \tag{6}$$

We normalize the values of the first form to be in the range $(0, 1)$ and after each derivative, we normalize again. This gives us a local feature form as a function of both scale and derivative order. This is similar to scale-space operators used in image processing. The problems are not directly correlated, however, because we assume our input surfaces are given in the same absolute scale, and do not need to compare signatures across multiple scales. Looking at derivatives of the function is useful because depending on how the inputs are generated, absolute magnitudes of the initial feature 0-form may not match, but it is still possible that the derivatives of this function match well. We now sample this function to get $k_{max}n_{max}$ feature forms:

$$\mathbf{h}_k^n = L^n\mathbf{g}_{t_{base}+\Delta tk}, \{0 \leq k < k_{max}, 0 \leq n < n_{max}\} \tag{7}$$

Taking higher order derivatives before significant diffusion results in very noisy values that are not useful for matching. Although these forms can contain unique information at various scales, in practice matching can be done with just one or two representative elements from this set. Let $\mathbf{c}_1, ..., \mathbf{c}_k$ denote the forms that we retain for matching. If we densely sample in k and do not look at any derivatives, then this is equivalent to comparing two points by integrating the difference between their value vs. scale curves. Figure 2 gives an example of the smoothed, higher order terms. Even without the oriented matching (described in the following section) the value vs. scale curves and its derivatives have good discriminatory capability (figure 3).

3.4 Grid Construction

The scalar values in $\mathbf{c}_i$ incorporate information from their localized neighborhood while remaining rotationally invariant. Many approaches to surface matching, such as SIFT features, patch features, and spin images, use rotationally invariant signatures for their final matching phase. This discards rotational relationships between local surface properties which can provide significant discriminatory power. [Gatzke 2005] explored rotational dependencies in signatures using curvature as the underlying feature form, with the rotational dependence collected into a geodesic fan. To aggregate the feature forms $\mathbf{c}_i$, we construct a local grid of resolution $D \times D$ and fixed side length around each triangle center $\mathbf{d}$. Triangles are a natural choice as a grid seed since points inside the triangle have both a well defined normal and zero Gaussian curvature. First, we choose an arbitrary local coordinate frame where $\hat{\mathbf{n}}$ is the triangle normal, $\hat{\mathbf{s}}$ is a random vector in the plane of the triangle and $\hat{\mathbf{t}} = \hat{\mathbf{s}} \times \hat{\mathbf{n}}$. We want to project each point $\mathbf{p}$ on the surface into the $\hat{\mathbf{s}}, \hat{\mathbf{t}}$ plane. Let (r, θ) be the geodesic coordinates of $\mathbf{p}$ relative to

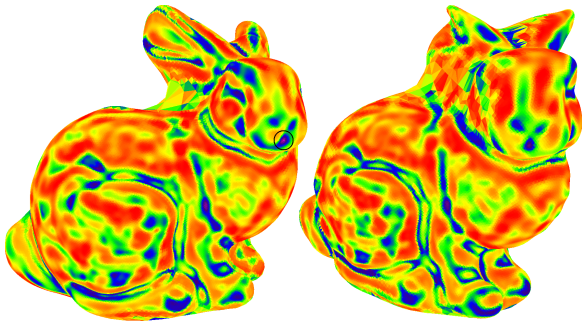

Figure 3: Integral between value vs. scale curve and its first three derivatives, as compared against the circled query point. The form and its derivatives were all assigned equal weight. Shown with a linear scale, violet is a perfect match, red is the worst match.

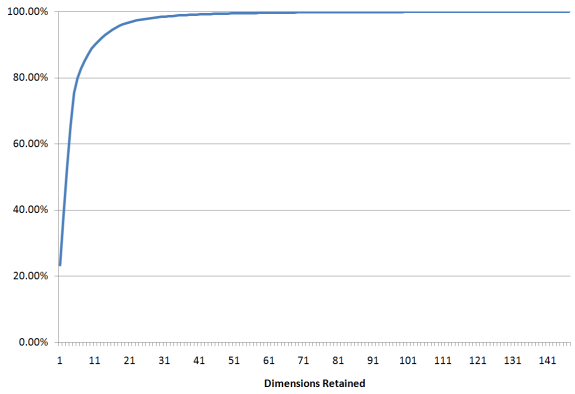

Figure 4: Percentage of total eigenvalue energy as a function of dimensions retained when performing PCA on three feature forms on a 7x7 grid for a total of 147 dimensions. Retaining 27 terms is sufficient to capture 98% of the energy.

the triangle center, where $\theta = 0$ corresponds to the direction of $\hat{\mathbf{s}}$. Consider these to be polar coordinates in the $\hat{\mathbf{s}}, \hat{\mathbf{t}}$ plane with the grid centered at the origin. We want to project the per-vertex values in $\mathbf{c}_i$ onto this grid; however, to avoid sampling issues due to non-uniform vertex distributions, we instead project points in our high resolution mesh sampling, $\mathbf{S_a}$. Since this set is sampled from the mesh surface, we have barycentric coordinates for each point and simply define the local feature value for each point in this set by interpolating the value from the vertices of its corresponding triangle. Let $G_{(i,j),\mu}$ and $G_{(i,j),\sigma}$ denote the mean and standard deviation of the points in $\mathbf{S_a}$ that project into grid cell (i, j). We form our final global descriptor by encoding each grid cell into two dimensions, $(G_{(i,j),\mu} + VG_{(i,j),\sigma}, G_{(i,j),\mu} - VG_{(i,j),\sigma})$, where V is the significance of the standard deviation matching (in all our examples, we use $V = 2$.) The union of all of these grid cells for each form $\mathbf{c}_i$ gives us the final oriented signature vector, with dimensionality $2kD^2$. As we increase grid resolution, or as we increase the number of feature forms retained for matching, many of these dimensions will become redundant. Since this will significantly increase the running time of the algorithm, we first perform principal component analysis on all signatures in the mesh, retaining a number of dimensions equal to 98% of the total energy (see figure 4).

4 Establishing Correspondences

4.1 Feature Matching

To compare the shape at two surface points, we compare every rotation of the oriented signature at one point against the signature of the other point at a fixed orientation and choose the best match. Computing this distance across all triangles is prohibitively expensive. To make this computation efficient for a large number of queries, we first restrict the problem to finding the triangle in mesh M_a that is the best candidate match for the query triangle on mesh M_b. We start by computing the signature at fixed orientation for every triangle in M_a and store this in an ϵ-approximate high dimensional nearest neighbor search structure. We then construct the signature for the query triangle starting with an arbitrary orientation vector $\hat{\mathbf{s}}$ and rotating the orientation vector by $\frac{2\pi}{d}$ each step. For each such orientation we query the nearest neighbor structure for the best match, and chose the best over all tested rotations. In total this is d queries on the NN structure with $Triangles(M_a)$ points. An alternative would be to store all possible rotations of each triangle in M_a in the NN structure, which would require 1 query on a NN structure with $Triangles(M_a)d$ points; however, in general the memory cost of this method is considerable. Since we expect the correspondence mapping to be highly coherent, we do not make a correspondence guess for every source triangle. Instead we use our low resolution mesh sampling, $\mathbf{S_b}$, as seeds and compute the most likely correspondence candidate on the other mesh for each such seed triangle. Although the distance between the two signatures serves as a good indication of how likely it is that the points correspond to each other, we accept a match for every seed point on both meshes and rely upon the subsequent stages to remove incorrect correspondences.

4.2 Pruning Correspondences

At this stage in the algorithm, every point admits a best fit candidate, regardless of how good this match it. A very simple technique is able to remove the large majority of these false correspondences. Suppose we determine that triangle T_a is the best candidate correspondence in M_a for the query triangle T_b in M_b. We then search M_b for the best match to the signature of T_a to get $T_{b,reflexive}$. Ideally, T_b and $T_{b,reflexive}$ would be the same triangle, and any significant geodesic distance between their two centers implies that the match is poor because they are not mutually best matches. We reject all correspondences for which this distance is greater than one-fourth the length of the grid used for signature construction. Denote the remaining set of correspondences by $Corr_{a,b}$. For a correspondence C, let C_a be the point in M_a, C_b be the matching point in M_b, and $C_{transfer}$ be the rotation matrix that transforms the local coordinate frame around C_a into the coordinate frame around C_b, which we determine from the oriented signature and triangle normal.

4.3 Correspondence Graph

False correspondences can still remain at this stage due to a combination of noise in the scan, symmetry on the underlying surface, and ghost surfaces created by the surface reconstruction. Generally, however, these spurious correspondences will not persist over any significant region, and we can use this to identify and remove these matches. Two correspondences with source points that are close in M_a should both also be close in M_b and their rigid transformations should agree. Given two correspondences C_1 and C_2, we can define an inconsistency metric by

$$Inconsistency(C_1, C_2) = \frac{\|C_{2,transfer}C_{1,a} - C_{1,b}\| + \|C_{1,transfer}C_{2,a} - C_{2,b}\|}{2} \tag{8}$$

This metric has units of distance, which is very useful for determining reasonable cutoff values. We want to accept sets of correspondences that are highly consistent over a large region. The Riemannian graph of a point set connects two points if either is in the set of k nearest neighbors of the other. Call $Riemannian_a(CorrGraph(Corr_{a,b}))$ the Riemannian graph with k = 16 constructed from the points C_a from each correspondence in $Corr_{a,b}$, and equivalently define $Riemannian_b$ constructed from the C_b's. Let $CorrGraph(Corr_{a,b})$ denote the graph of the correspondences where we place an edge between two correspondences C_1 and C_2 if $C_{1,a}$ and $C_{2,a}$ are neighbors in $Riemannian_a(CorrGraph(Corr_{a,b}))$ and $C_{1,b}$ and $C_{2,b}$ are neighbors in $Riemannian_b(CorrGraph(Corr_{a,b}))$.

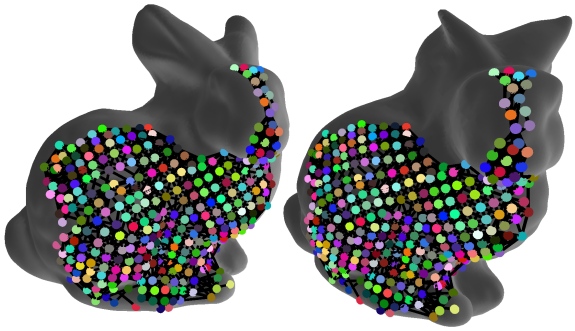

Figure 5: Largest connected component when computing correspondences on the scans from figure 1. The next largest component has 3 correspondences, and only 7 components with more than one correspondence were found. Decreasing the maximum distance permitted between the seed and reflexive triangle, or the maximum inconsistency allowed in the correspondence graph, could further remove inaccuracies at the cost of fewer correspondences reported, if desired.

Each edge in this graph has an associated consistency measure. We discard all edges with an inconsistency greater than one fourth the distance between the two corresponding points in either M_a or M_b. We then extract connected components from $CorrGraph(Corr_{a,b})$. If the surface matching is good, the largest such connected component will be significantly larger than all others, which rarely contain more than one or two vertices. If the surface matching is poor (for example, two range scans that do not overlap the underlying surface) the largest connected component will generally be very small. In the case of rigid surface matching, generally only the largest connected component is important. However, when doing non-rigid matching, there may be several valid connected components due to occlusions in the scans or significant deformations. Because two correspondences only have to locally agree to be connected, it is still possible for a single component to contain significant non-rigid deformations if locally the deformations are sufficiently small.

5 Results

The final correspondences computed for two scans of the Stanford bunny with significant overlap are shown in figure 5 while figure 6 shows two scans from opposite sides. Both these results use three feature forms and a 7x7 grid for a total of 147 dimensions before PCA. The non-overlapping case produced almost no correspondences while the overlapping scans produced a single large correspondence that covers virtually the entire shared region. The overlapping parts of the scan that do not produce correspondences are generally near the boundary, and are missed due to the diffusion process, which incorporates values from regions not part of the original surface that are produced by the Poisson reconstruction, and due to the grid radius which also incorporates information from these reconstructed regions.

6 Conclusions and Future Work

A significant advantage of the method presented in this paper is that it can produce dense correspondences. Many techniques, such as patch features and spin images, cannot produce dense matches or incur a high degree of uncertainty if they do so. Furthermore, the fact that the correspondences are oriented significantly increases the amount of information conveyed in the mapping compared to a point only mapping, which would require inferring changes in the local coordinate frame by looking at a dense mapping of points and would suffer greatly in the presence of noise in the mapping. These properties make it well suited for computing correspondences in non-rigid situations. The two main contributions, a local point signature that is invariant to rigid rotations, and a robust method for computing well pruned dense correspondences given such a local signature, are independent of each other and the signatures can be used for applications beyond correspondence. The technique is also

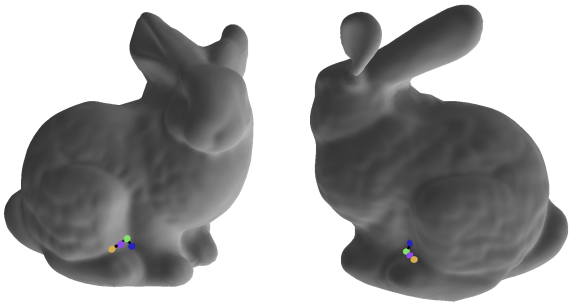

Figure 6: Largest connected component when computing correspondences on a scan of the front and back of the Stanford bunny. In total only 9 components with more than one correspondence were found.

not iterative and does not require a reasonable initial guess as does the iterative closest point algorithm and its variants. However, similar to spin images, the relatively high dimensionality of the signature means that finding exact nearest neighbors incurs substantial computational cost. The signature generation also relies on an underlying base descriptor, such as color from an associated image of the surface or PCA eigenvalues from a local neighborhood, and its basic matching capability derives from this value being similar between inputs. While the algorithm is relatively robust to noise due to the diffusion process and best candidate search, when the noise magnitude becomes comparable to the magnitude of surface features, the performance of the algorithm can degrade noticeably. The algorithm also cannot function well in the absence of localized features or when these features cannot be relied upon (such as a T-shirt undergoing significant folding and creasing between scans.) Except in the case of exceptional symmetry in the underlying shape, it significantly favors returning almost no correspondences over producing false positives. The computed correspondences also do not take the mapping of neighboring correspondences into account except during the pruning phase. Using information from the correspondence neighborhood in conjunction with the signature metric to iteratively improve the final mapping could yield significant improvements. Likewise, determining good feature scales and grid sizes when performing the matching remains a significant open question.

7 References