Isometric Registration of Ambiguous and Partial Data

Art Tevs
Max-Planck Institute Informatik
tevs@mpi-inf.mpg.de

Martin Bokeloh
WSI/GRIS, University of Tübingen
bokeloh@gris.uni-tuebingen.de

Michael Wand
Max-Planck Institute Informatik
mwand@mpi-inf.mpg.de

Andreas Schilling
WSI/GRIS, University of Tübingen
schilling@uni-tuebingen.de

Hans-Peter Seidel
Max-Planck Institute Informatik
hpseidel@mpi-inf.mpg.de

Abstract

This paper introduces a new shape matching algorithm for computing correspondences between 3D surfaces that have undergone (approximately) isometric deformations. The new approach makes two main contributions: First, the algorithm is, unlike previous work, robust to "topological noise" such as large holes or "false connections", which is both observed frequently in real-world scanner data. Second, our algorithm samples the space of feasible solutions such that uncertainty in matching can be detected explicitly. We employ a novel randomized feature matching algorithm in order to find robust subsets of geodesics to verify isometric consistency. The paper shows shape matching results for real world and synthetic data sets that could not be handled using previous deformable matching algorithms.

1. Introduction

Deformable shape matching has recently gained a lot of interest. One application of deformable shape matching is in improving the registration of 3D scans [9]: Even expensive 3D scanners suffer from minor calibration problems that lead to alignment problems if large models are acquired at a high spatial resolution. This can be compensated by allowing for small global deformations. An application area that requires handling much larger deformations stably is the acquisition of deformable objects. For example, scanning of humans, animals or other living beings usually requires multiple scanning passes from different perspectives during which the scanned subject will inevitably move in a non-rigid fashion. Hence, assembling complete, high-resolution scans requires deformable shape matching. A related application area is animation scanning [20, 23, 25, 28, 30]. In this line of work, which has received quite some attention recently, the goal is to capture the dynamics of a moving object in real-time. Again, it is necessary to construct a composite object from several, probably strongly deformed poses with significant deformation. The available data in each frame of an acquired animation is usually incomplete, as typically being measured from a sparse set of views only. Therefore, the matching algorithm needs to be able to deal with partial data, matching portions of the surface with several large holes.

Several algorithms have been proposed to solve this deformable matching problem in a time-sequence manner, where only temporally adjacent frames are combined under the assumption of spatial coherence, using deformable variants of the ICP algorithm, such as [4, 14, 25, 28]. These techniques yield good results for small inter-frame deformations. However, they are rather unstable under fast movements of the scanned object that lead to substantial differences in pose. Matching objects with strong deformations is easy if a set of guiding markers is available that specify the rough pose of the object. Using this information (often provided by manual labeling [3, 4, 22]), the shapes are roughly prealigned and afterwards a fine-scale alignment is performed by deformable ICP. Recently, a number of fully automatic techniques have been proposed for *global*, pose independent deformable matching by computing such marker sets automatically. The key idea for these techniques is to preserve geodesic ("intrinsic") distances, assuming that the deformation of the object is approximately isometric [1, 5, 7, 10, 15, 24].

In this paper, we extend this line of work. Our goal is also to find a suitable set of markers that could be used to guide a locally convergent deformable ICP algorithm. This is done fully automatically, without user intervention and without assumptions on pose or deformation. Our only requirement is that the unknown deformation is approximately isometric, i.e., roughly preserves distances on the surface. The novel aspect of our work is that we propose a technique that still works under *topological noise*. We make three main contributions:

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

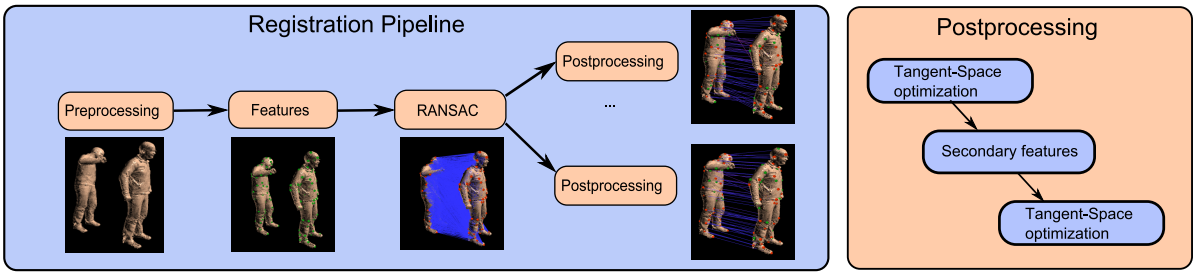

Figure 1: Pipeline overview of our algorithm. A pair of input point clouds is first preprocessed Sect. 3.1. Afterwards a set of feature points and a superset of all correspondences is computed, Sect. 3. Afterwards a RANSAC-like search of candidate solutions is performed, Sect. 3.3. And finally each solution is being postprocessed, Sect. 3.5 and 3.7.

- We propose a randomized, RANSAC-like matching strategy that, unlike previous techniques, simultaneously estimates the correspondences and the validating geodesics in an outlier-robust way.
- We employ a new tangent-space optimization algorithm to optimize the placement of feature points for maximum isometric matching, yielding better result for noisy feature positions.
- We sample the space of plausible matches, which gives us the ability to explicitly examine matching alternatives. We consider this an important building block for fully automatic matching of complex models from several pieces.

We also consider the problem of upsampling sparse initial correspondence sets to denser sets [1, 15], where our technique again has the advantage of being resistant to topological noise. We apply our matching algorithm to a number of data sets and show reliable matching results for general pose and topologically distorted data with holes and false connections.

2. Related Work

Deformable shape matching can be classified in local and global matching strategies. Local matching is based on deformable variants of ICP [4, 18, 28], where point-to-surface distances are minimized under the regularizing assumption of elasticity. Alternative formulations that have been proposed include optical-flow-like correspondence propagation [20, 27] for densely sampled time sequences and Laplacian diffusion [1] for correspondence interpolation. Li et al. propose a semi-global matching technique that is more robust in convergence by numerically optimizing matching weights [18]. Bronstein et al. [7] examine the problem of embedding surfaces into each other isometrically using a numerical optimization scheme (generalized multi-dimensional scaling). Our numerical refinement scheme for feature positions uses a similar objective function. Anguelov et al. [5] use loopy belief propagation to solve a graph matching problem. This idea is extended in [24], using only local neighborhoods, which allows for some resistance to topological changes. A simple and very effective matching algorithm for pairwise constraints has been proposed in [17], relaxing quadratic assignment to an eigenvalue problem. This algorithm can be applied to deformable matching using feature detection to define keypoints and the preservation of geodesic distances between pairs of features as pairwise validation criterion. This approach is used for example in [1, 6, 15] for global shape matching and is currently probably the most frequently used and state-of-the-art technique. The main drawback of this algorithm is that the pairwise validation criterion (sets of geodesic distances) have to be specified upfront. Therefore, a small number of topological problems that reroute some of the geodesics can drastically impact on the matching results. Other deformable matching techniques include optimization with deterministic annealing [11] and Chang et al.'s technique [10] based on graph cuts. Zhang et al. [29] sample assignments of a sparse set of extremity feature. A similar idea is used for sequence merging in [26]. Automatic insertion of additional correspondences based on landmark features has been proposed to us by L. Guibas [personal communication], motivated by earlier work on landmark for routing in sensor networks [12]. It has been examined recently more in detail by Huang et al. [15], propagating denser correspondences by geodesic landmark coordinates, and Ahmed et al. [1], who employ Laplacian diffusion of correspondence information. A related proposal for dealing with this problem has been made by Bronstein et al. [8], who trade-off Euclidean and intrinsic distances for validating matches. They show significant improvements over intrinsic-only matching criteria but the technique is not able to handle general cases of strong deformations, where Euclidean matching becomes very unreliable.

3. Isometric Matching

We start our matching algorithm by computing correspondence candidates by feature matching. This gives us a superset of correspondences from which we have to ex-

tract a correct subset. We also compute geodesic distances between all feature points, which gives us a superset of geodesics where some of those might actually be incorrect and give false cues for validating the correctness of the correspondences. We are now looking for a subgraph of consistent correspondences and geodesics that is maximal in the sense that no edges can be added without exceeding an error threshold. Out of the many different such solutions that may exist, we will prefer large solutions with many correspondences, each validated by many pairs of geodesic distances, as these are less likely to be spurious matches. In order to compute such a solution, we use a RANSAC-like algorithm that randomly samples the solution space. For efficiency reason, we will bias the random search towards promising matching candidates by employing a suitable importance function. In order to perform this algorithm reliably, we need a criterion for the reliability of verifying geodesics, which we develop subsequently. Next, we employ a tangent-space optimization technique to compute an optimal placement of the features. After having obtained a good set of feature matches, we extend the correspondences to so far unlabeled space by inserting new "secondary" features derived from distances to matching correspondences. Finally, this dense feature set is also fine tuned by applying tangent-space optimization again. In an outer loop, we execute the whole matching algorithm repeatedly in order to find matching alternatives which are ranked by how well they explain the input data and output to the user. In the following, we discuss all of these steps subsequently.

3.1. Input Data and Preprocessing

We directly work on the point-based representation as provided by a 3D scanning device. We expect two sets of 3D points $\mathcal{X}^{(j)} = \{\mathbf{x}_i^{(j)}\}, i \in \{1..n_j\}, j \in \{0, 1\}$ as input. These two point sets are samples of two (unknown) smooth manifolds $\mathcal{S}^{(j)}$. We first compute normals from the k-nearest neighbors of each data point ($k = 20$) using PCA. Additionally, we also compute a "topology graph" for each point cloud. This graph just connects every point to its k-nearest neighbors (again, we typically use $k = 20$) and approximates the local connectivity of the surface. Please note that this graph does not need to form a valid triangle mesh. We will use the graph distance in this graph (using Dijkstra's algorithm) as an approximation for geodesic distances in the sampled manifold, which is the standard approach for fast estimation of geodesics.

3.2. Feature Matching

In the next step, we compute a candidate set of surface feature points. In principal, any surface feature matching technique can be employed at this point [13, 16, 19]. In our work, we use the slippage feature matching technique [6]. Slippage features detect keypoints by maximizing the stability of the auto-alignment of local surface pieces in scale space, which leads to a large number of stable feature points. In the following, we will refer to the features on surface j as $\mathbf{p}_i^{(j)}, i \in \{1, ..., k_j\}$. Given a set of feature points, we next compute a local descriptor for small circular neighborhoods of each feature. We use a rather simple descriptor that just computes a histogram of mean curvature in this region. Mean curvature is computed from a quadratic moving-least-squares surface approximation (see [6] for details). Having computed feature points and descriptors, we build an initial matching graph: We connect all features on $\mathcal{S}^{(0)}$ and $\mathcal{S}^{(1)}$ with *correspondence edges* for which the descriptors are similar up to at least a user defined threshold (we choose a conservative threshold that leads to many false positives, as we will subsequently filter the graph further). Additionally, we also connect all pairs of features $\{\mathbf{p}_i^{(0)}\}_i$ in shape 0 and all pairs of features $\{\mathbf{p}_i^{(1)}\}_i$ in shape 1 with *validation edges* and compute the approximate geodesic distance between these pairs. Each validation edge is tagged with this distance.

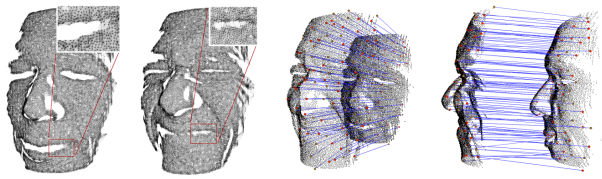

Figure 2: Faces data set and its registration. The dataset represents a hard case with different topology due to the connections in the mouth region.

3.3. RANSAC Subgraph Extraction

Given the input candidate graph, we now extract a consistent subset using a RANSAC-like randomized sampling algorithm. Naively, we could enumerate all possible subgraphs of our candidate set and evaluate how well it explains the data. However, this would lead to costs of $O(2^N)$ where N is the overall number of validation and correspondence edges. By random sampling, we can examine the same search space in expected time $O(N \cdot 2^N)$ [21], which is slightly worse. However, by augmenting the sampling density, we can obtain a more efficient solution. Obviously, it is not reasonable to try matches that are very unlikely to be correct. Therefore, we will employ importance sampling, which distorts our random sampling density in a way to yield promising matching candidates with larger probability. This will drastically reduce the expected time required to find a good match. We can further motivate the efficiency of this scheme using a heuristic argument: For the first match, we can only rely on the rather noisy descriptor matches as importance function. These matches typically have outlier rates of 80%, i.e. 4 out of 5 matches are wrong. However, once we have found at least 2-3 matches, this establishes a local coordinate frame in terms of geodesic dis-

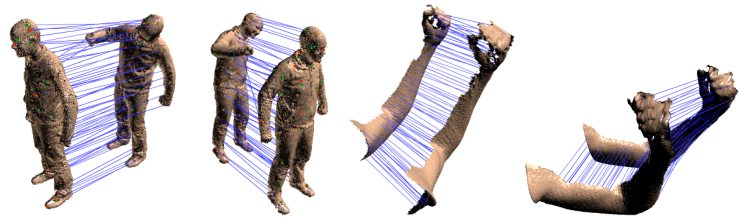

Figure 3: Matching results: (Left) guy data set 75 features (35 secondary features) were registered. (Right) arm-dataset registered 84 feature (56 secondary features).

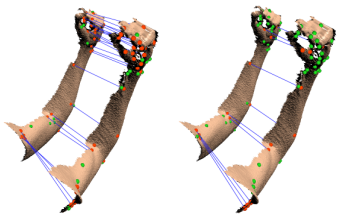

Figure 4: Arm data set matched with (left) our algorithm and (right) spectral matching.

tances, so that it becomes much easier to judge whether new correspondence candidates are correct. From this perspective, we only need a small number of random guesses to find a candidate set that bootstraps the matching process, rather than an exponential number in N.

In order to implement this strategy concretely, we employ the following algorithm: We start by selecting a random correspondence $m_{i,j} = (\mathbf{p}_i^{(0)}, \mathbf{p}_j^{(1)})$ according to an importance sampling density $p(m_{i,j})$. Initially, this density is chosen to be proportional to the descriptor matching scores. Then we iteratively add more correspondences using an augmented sampling density that takes the geodesic distances to existing features into account. These distances will not be exact because of slightly non-isometric deformations, as well as noise in the feature location and distance estimation process. For simplicity, we assume that the aggregate of these errors, for a *correct* match, yields a Gaussian distribution. The standard deviation of this distribution could be calibrated by user-labeled example matches from the data source considered; we currently set this parameter manually. If we already have several correspondences in our candidate sets for the correct subgraph, we have to check multiple geodesics. We now assume that the error affecting the geodesic lengths is independent for each geodesic. While this is certainly not exactly the case, it is a reasonable approximation in practice. In this model, we can compute the probability of length distortion. Let l_k denote the actual geodesic distance observed and $l_k^{(0)}$ the correct geodesic distance, the probability density for this deviation, given the match $m_{i,j}$ is correct, is given by:

$$p(l_1, ..., l_{n(i,j)}|m_{i,j}) = \frac{1}{\sigma^m (2\pi)^{m/2}} \prod_{k=1}^{m} exp\left(-\frac{\left(l_k - l_k^{(0)}\right)^2}{2\sigma^2}\right) \quad (1)$$

Now we can employ Bayes rule to compute the probability of match $m_{i,j}$ being correct, given we know how well geodesic distances are preserved:

$$p\left(m_{i,j}|l_1, ..., l_{n(i,j)}\right) = \frac{p\left(l_1, ..., l_{n(i,j)}|m_{i,j}\right) p\left(m_{i,j}\right)}{\sum_{i,j} p\left(l_1, ..., l_{n(i,j)}\right) p\left(m_{i,j}\right)} \quad (2)$$

Here, $p(m_{i,j})$ is the prior probability of match $m_{i,j}$ being correct, which in our case is given by the descriptor matching. This formula can directly be used to choose the next match to be added to our current subgraph: We compute the matching likelihood, given by this formula, and use the corresponding probability density for importance sampling. For importance sampling, we generally exclude any matches that receive a very low matching probability (typically, below 1%). Therefore, the iteration terminates automatically when no more reasonable matches are available. As an alternative to pure importance sampling, we can also use only the best (maximum a posteriori) match once a sufficient number (typically 3-5) of base correspondences have been sampled. In our experiments, this leads to a better results in comparison to the purely random algorithm.

3.4. Handling Topological Noise

The algorithm outlined above yields very good results, comparable to or even better than state-of-the-art matching algorithms such as spectral geodesic matching [15, 17] (see Figure 7). However, this may fail once the geodesics become unreliable. As mentioned in the conclusions of [15], one needs to be able to handle geodesics in an outlier robust way, which is hard to incorporate in standard techniques such as pairwise spectral validation. For our RANSAC-like matching algorithm, however, this is rather easy to achieve: In a plane, two different points are sufficient to find unique geodesic coordinates for all other points up to mirroring along the line defined by the coordinate points, and three non-collinear points define a unique barycentric coordinate system. On curved manifolds, the situation might be more complex. However, in practical cases, it is highly unlikely to have two different points with the same geodesic distances to a larger number of feature points (say, to 10 different points). Therefore, we do not need to guarantee full geodesic consistency but it is sufficient to have a large enough witness set that proves the correct match, while further geodesics inconsistencies can be regarded as outliers due to topological noise. We now augment our RANSAC matching loop as follows: In computing the matching probabilities (1), we do not use all geodesics but determine the subset of geodesics that does not show too large deviations in distance across shapes and use only this subset for validation. In order to make this reliably, we demand to have at

least k such geodesics, with k typically in the range of 5-10. This means that out of n geodesics, $n-k$ are allowed to be outliers, we only need a minimal proof of correctness.

3.5. Tangent-space optimization

Feature positions are not exact as they rely on the ability of the feature detection algorithm to find well-defined spots. We usually cannot effort to use a very firm threshold on the well-constraintness of the feature position because this would remove too many features, limiting the scope of the matching. In addition, we may have wrong matches to actually different but nearby features that still meet our geodesic validation criterion. By moving these features slightly we can obtain a much better matching. Assume we are given k reference points and one more point $\mathbf{x} \in \mathcal{S}$ with geodesic distances $\mathbf{l} = (l_1, ..., l_k)$. At $\mathbf{x}$, we form a two-dimensional tangent space coordinate system of orthogonal vectors $(\mathbf{u}, \mathbf{v})$ that are orthogonal to the surface normal $\mathbf{n}(\mathbf{x})$. Let $\mathbf{x}^{(uv)}$ be the coordinates of $\mathbf{x}$ in tangent space. We now consider the partial derivatives of the position of $\mathbf{x}^{(uv)}$ by the length of the geodesics. Up to first order, we obtain:

$$\triangle \mathbf{x}^{(uv)}(l_1, ..., l_k) \doteq \underbrace{\left(\frac{\partial \mathbf{x}^{(uv)}}{\partial l_1} \cdots \frac{\partial \mathbf{x}^{(uv)}}{\partial l_k} \right)}_{=:\nabla \mathbf{l}} \triangle \mathbf{l} \qquad (3)$$

where $\triangle \mathbf{l}$ is the displacement in geodesic coordinates. We get a linear map $\nabla \mathbf{l}$ that transforms small displacements in geodesic coordinates into displacements in spatial coordinates. In order to compute the gradients $\partial \mathbf{x}^{(uv)} / \partial l_i$ we consider a spherical neighborhood $S(\mathbf{x})$ of points and compute geodesic distances to the reference point for all points in the point cloud. This comes at no additional cost, as this has to be done during the Dijkstra algorithm anyway. Afterwards, we compute a least-squares fit of a linear model to the resulting distance values (with Gaussian weighting window with standard deviation proportional to the radius of $S(\mathbf{x})$). This means, we compute the coefficients a_0, a_1, b of a linear model $\langle \mathbf{a}, \triangle \mathbf{x} \rangle + b$ that fits the observed distances best in a weighted-least-squares sense. The vector $\mathbf{a}$ then yields our gradient approximate in tangent space. Using this first order approximation, we can setup a quadratic objective function that locally describes how well the length of matching geodesics is balanced. We then optimize this function, move the points in tangent direction, project them back on the manifold using an MLS surface approximation [2], and recompute new estimates for geodesic distances and their gradients in every step, leading to a fairly efficient Gauss-Newton-type iteration.

We obtain the quadratic objective function by expressing the change of length of the geodesics in terms of gradients of the geodesics with respect to the point position. In the following, we use $l_{i,j}^{(k)}$ to denote the intrinsic distance between two feature points $\mathbf{p}_i^{(k)}$ and $\mathbf{p}_j^{(k)}$ on surface $k \in \{0, 1\}$.

Figure 5: Error of approximated intrinsic distances. The image shows the *maximum* error over all geodesics between each point and all other points of the point cloud. *Horse*: 8431 original points, 1355 sample points - max error is 5% of the bounding box size. *dragon*: 20002 original points, 1598 sample-points - max error is 10%.

Correspondingly, we use $\mathbf{g}_{i,j}^{(k)} := \partial \mathbf{p}_i^{(k)} / \partial l_{i,j}^{(k)}$ to denote the gradient of the geodesic distance with respect to the tangent space ($\mathbf{uv}$-) coordinates of a feature point $\mathbf{p}_i^{(k)}$. The notation means that the gradient is measured at feature i on surface k for a geodesic that connects feature i to feature j. This yields the following objective function:

$$\arg\min_{\triangle p_i \in \mathbb{R}^2} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \left(l_{i,j}^{(0)} + g_{i,j}^{(0)} \cdot \triangle p_i^{(0)} + g_{j,i}^{(0)} \cdot \triangle p_j^{(0)} - l_{i,j}^{(1)} \right)^2$$

This is a quadratic objective function in the unknown tangential displacements $\triangle p_i^{(0)}$. We set the gradient of this energy to zero and solve the resulting linear system using an SVD, which is robust to degenerate cases (numerically small absolute singular values are not inverted). We then move the feature points tangentially according to the computed displacements in three-space. Afterwards, we project each point back on a surface approximation obtained from moving least square fit (with quadratic basis functions and Gaussian weights). Then, geodesics are recomputed and the scheme is iterated until it converges, i.e., only small changes in energy occur. We move the point by the surface sample spacing ϵ_{sampl} in order to prevent missing the surface in the projection step. In oder to make the scheme symmetric, we alternate between optimizing the feature positions on surface $\mathcal{S}^{(0)}$ and $\mathcal{S}^{(1)}$ in each iteration. In practice, this symmetric approach shows to be significantly more robust and accurate than a one-sided optimization, moving feature points only on one surface.

3.6. Approximation of Geodesics

Each iteration of the tangent-space optimization step as described in the previous section requires a recomputation of intrinsic distances $l_{i,j}^{(k)}$ and corresponding gradients $\mathbf{g}_{i,j}^{(k)}$ that involves running Dijkstra's algorithm for each feature point, which is quite costly. In order to solve this problem we use an algorithm to precompute an approximation to the geodesic distance and the corresponding gradients between any two points on the point cloud surface: First, we compute sampling points $\mathbf{d}_n$ from the original point cloud. We

downsample the point cloud in respect to the curvature by a Poisson-disc sampler with radius inversely proportional to the curvature. For the downsampling factor we have a trade-off between the absolute error and preprocessing time we are willing to tolerate. Figure 5 shows the absolute error achieved with typical parameter settings.

Having the data points we now compute graph distances between the sample points using the original, full resolution graph for distance estimation. Next, we interpolate intrinsic distance $l_{i,j}$ and gradient $\mathbf{g}_{i,j}$ between any points $\mathbf{p}_i$ and $\mathbf{p}_j$ on the surface. First we determine k nearest data points around both points $\mathbf{p}_i$ and $\mathbf{p}_j$ respectively where k is usually 4. Next, we place Gaussian basis functions around each sample point with radius (standard deviation) proportional to the point spacing and obtain the interpolated distance by a partition-of-unity interpolation: For each source point, we compute a partition-of-unity interpolation of all distances to the destination and interpolate these again using a second partition-of-unity interpolation in the destination domain.

3.7. Inserting Dense Secondary Features

The robust criterion for geodesic validation can also be employed to insert new secondary feature points $\mathcal{S}_P$ with correspondences into our model. This is desirable as the feature detection stage often fails to yield a sufficient number of feature points to cover the surface densely enough. Inserting new correspondences is easy: We first pick a random surface point out of the sample points $\mathbf{d}$ as described in previous subsection. Then, we go through all points of the target surface and pick the best matching one. We apply exactly the same matching criterion as for regular features described previously: We require a minimum number of geodesic distances to be correct within tolerance and that this subset is stable. We add the best matching of those correspondences to our correspondence set. This is iterated until all points are taken or no more reliable matches are found. In order to speed up this brute force search algorithm, we try to reduce the search area: First, we find all already matched features within a small neighborhood of $r \cdot \epsilon_{sampl}$ of the secondary feature point (typically: $r = 8$). We compute the Euclidean distances and perform a partition-of-unity interpolation of the corresponding points, interpolating the target positions in Euclidean space, which we then project back on the target surface. We now examine all target points within distance $r \cdot \epsilon_{sampl}$ using a breadth-first search on the topology graph. Only if this search yields no result, we fall back to the expensive brute force search. In practice, this yields a substantial speed-up.

3.8. Ranking of Matching Alternatives

The algorithm described so far will produce a single partial isometric match of the two input surfaces. In case of multiple valid solutions, it is a matter of chance which solution will be output. For equally plausible results (similarly isometric, same amount of features and area covered), the algorithm will output all these results with a significant probability. This randomization is actually useful to gain more information about the matching of the two shapes involved. We run the algorithm several times in an outer loop: Instead of using only the best initial feature match from the inner RANSAC loop, we keep the k-best matches (typically $k = 10$). We run the whole pipeline for the first match. For the next best feature match, we then determine whether it is contained in the previous solution. The problem here is that feature positions might be different so that we cannot directly compare the results. Therefore, we interpolate the matching results using a partition-of-unity Gaussian interpolation with window radius $2\epsilon_{sampl}$ and compare the obtained interpolated correspondences with the feature correspondences. Points without feature correspondences in their vicinity are treated as unmatched area and are ignored in the comparison. Only if there is a substantial disagreement, we compute a new solution from the initial feature match. Otherwise, we dismiss the match and go on examining the next best. This is repeated until all matches have been compared. In the end, we rank the results by counting the number of overall features (initial and secondary features) matched, which corresponds to matched area, multiplied by the number of geodesics that support each of the validated features. If the numbers are equal, the error in the deviation of geodesic distances is used as secondary sorting criterion. This output is a sampled description of the matching ambiguity present for the two shapes, with the "most likely" reconstruction listed first.

4. Implementation and Evaluation

We tested our algorithm on an Intel Core2 CPU with 2GHz and 2GB RAM with different synthetic and real world data sets. Table 1 summarizes the timings and statistics.

Synthetic data: First, we apply our algorithm to two well known 3D models, horse and dragon, which we sample synthetically. Figure 6 shows the matching results along with a comparison to ground truth. The horse data set is particularly hard to match because of the skinny and mostly featureless legs. In addition, the model is perfectly symmetric under left/right-mirroring in its rest pose. Because of the larger similarity between the right and left legs of the horse, our algorithm actually prefers the mirrored solution as best match. Figure 7 shows a registration of an artificial data set. The two frames of the data set are equal except of the topological holes added in the second frame. We were able to match more features enabling the outlier detection than with spectral matching [17] or with a simple RANSAC matching. For a fair comparison, matching was performed without tangent-space optimization and with no additional

data set	iterations	$\lvert\mathcal{S}^{(0)}\rvert \cdot \lvert\mathcal{S}^{(1)}\rvert$	sample pts	precomp.	RANSAC	registr.	$\lvert\mathcal{M}\rvert$	score	max error	mean error
dragon	500	4760	3196	40m13s	20.2s	7m2s	95	7.43	0.11	0.015
horse	100	7740	1233	9m20s	36.5s	11m49s	100	6.95	n.a.	n.a.
guy	500	2646	868	9m4s	36.1s	3m40s	75	5.01	n.a.	n.a.
arm	500	1440	1974	16m13s	12.5s	4m38s	84	0.67	n.a.	n.a.
face	100	7238	1920	17m30s	68.3s	8m10s	103	4.82	n.a.	n.a.

Table 1: Numerical results: test data set, number of RANSAC iterations, number of candidate correspondences, sample points for geodesics approximation, precomputation time, RANSAC and overall matching time, matching score [log-likelihood in Eq. 1], maximum and average error w.r.t. ground truth [where available]). Please note that the actual RANSAC search is quite fast; most of the time is spend in precomputing geodesics and tangent space optimization.

secondary features.

3D scanner data: We examine three real world 3D scan data sets (original raw data up to downsampling): "guy", "arm", and "face". All three suffer from both geometrical and topological noise problems. Figure 3 shows the result for the arm data set. Please note the big acquisition hole on the palm as well as the fingers touching in only one scan. Our algorithm was able to register 84 features. Figure 4 shows a side-by-side comparison of our algorithm with spectral matching, which does not capture features in regions of topological variation. To compare both algorithms neither tangent space optimization was performed nor secondary features were added. Results for the face data sets are shown in Figure 2. Again, in the mouth and eye regions, false topological connections occur. We were able to register 103 features (49 secondary features) in around 8 minutes. The example shows one limitation that our technique shares with all isometry-based matching techniques: The portion of the face above the upper lip is not deforming isometrically (at least not as portrait at scanner resolution), so that the upper lip matches a bit too low. As this does not appear as a spurious outlier, but a consistent shift, the registration does not yield the intuitively expected result. For the rest of the face, we obtain reliable results. Figure 8 shows results for partial matching with ambiguities - we match the index finger of a scanned hand to the full hand model. Among the 5 best matches are matches to two different fingers. See additional material for more extensive results.

5. Conclusions and Future Work

We presented a global deformable matching approach based on a novel RANSAC-like randomized sampling algorithm. The algorithm works for general geometric data sets solely assuming approximately isometric deformations. The algorithm is robust to topological noise and unlike previously known techniques, our approach is able to output matching alternatives by sampling the space of plausible solutions. This might be an important tool in multi part matching situations with ambiguous pairwise matches, such as animation sequence reconstruction. In future work, we would like to examine applications of our technique in that direction. In addition, we would also like to improve the approximation of geodesics both in terms of speed and stronger approximation guarantees than just the trivial bound of sample spacing resolution.

References

[1] N. Ahmed, C. Theobalt, C. Rossl, S. Thrun, and H.-P. Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[2] M. Alexa, J. Beht, D. Cohen-Or, S. Fleishman, D. Levin, and C. Silva. Computing and rendering point set surfaces. *IEEE Trans. Visualization and Comp. Graphics*, 9(1):315, 2003.

[3] B. Allen, B. Curless, and Z. Popović. Articulated body deformation from range scan data. *ACM Trans. Graph.*, 21(3):612–619, 2002.

[4] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.

[5] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*, 2004.

[6] M. Bokeloh, A. Berner, M. Wand, A. Schilling, and H.-P. Seidel. Slippage features. Technical Report WSI-2008-03, University of Tübingen, WSI/GRIS, 2008.

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS)*, 103(5):1168–1172, 2006.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Topology-invariant similarity of nonrigid shapes. *Intl. Journal of Computer Vision (IJCV)*, to appear.

[9] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-d scans. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[10] W. Chang and M. Zwicker. Automatic Registration for Articulated Shapes. *Computer Graphics Forum (Proceedings of SGP 2008)*, 27(5), 2008.

[11] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, 89(2-3):114–141, 2003.

[12] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang. Glider: Gradient landmark-based distributed routing for sensor networks. In *24th Conference of the IEEE Communications Society (InfoCom)*, 2005.

[13] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on Geometry Processing*, pages 197–206, 2005.

[14] D. Häehnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2003.

[15] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proc. SGP 2008)*, 27(5), 2008.

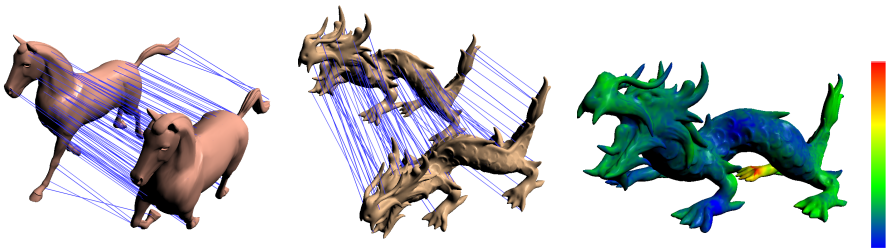

Figure 6: Synthetic data sets horse and dragon. (Left) 100 correspondences while matching the horse data set were found. (Right) dragon data set matched with 95 features (48 secondary) with a maximal error (red) of 11%.

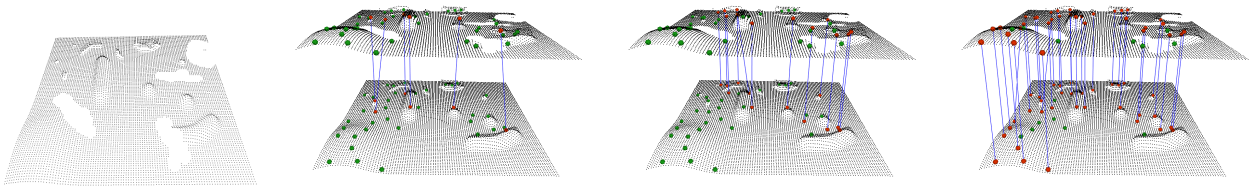

Figure 7: Synthetic dataset. Last three screen shots shows the spectral matching algorithm (6 matches), our algorithm without outlier detection (12 matches) and with outlier detection enabled (33 matches) respectively. No optimization step and no secondary features were added. $|\mathcal{S}^{(0)}| \times |\mathcal{S}^{(1)}| = 1400$

Figure 8: Five best results of a registration of the index finger and the whole hand. Due to symmetrical structure there exists ambiguous solutions which were found with our algorithm.

[16] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. Graphics*, 25(3):569–578, 2006.

[17] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference of Computer Vision (ICCV)*, volume 2, pages 1482–1489, October 2005.

[18] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP 2008)*, 27(5), 2008.

[19] X. Li and I. Guskov. Multiscale features for approximate alignment of point-based surfaces. In *Symposium on Geometry Processing*, pages 217–226, 2005.

[20] N. J. Mitra, S. Flory, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing*, pages 173–182, 2007.

[21] R. Motwani and R. Raghavan. *Randomized Algoirthms*. Cambridge University Press, 1995.

[22] M. Pauly, N. Mitra, J. Giesen, M. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Proc. Symp. Geometry Processing*, 2005.

[23] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)*, 2008.

[24] J. Starck and A. Hilton. Correspondence labelling for wide-timeframe free-form surface matching. In *Proc. Int. Conf. Computer Vision*, 2007.

[25] J. Süssmuth, M. Winter, and G. Greiner. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proceedings of SGP 2008)*, 27(5):1469–1476, 2008.

[26] K. Varanasi, A. Zaharescu, E. Boyer, and R. Horaud. Temporal surface tracking using mesh evolution. In *Proc. Europ. Conf. Computer Vision (ECCV)*, 2008.

[27] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *Proceedings of the 7th International Conference on Computer Vision*, volume 2, pages 722 – 729, September 1999.

[28] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[29] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-drive shape correspondence. *Computer Graphics Forum (Special Issue of Symposium on Geometry Processing 2008)*, 27(5):1431–1439, 2008.

[30] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime faces: high resolution capture for modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 548–558, New York, NY, USA, 2004. ACM.