Non-Rigid Registration Under Isometric Deformations

[1,2]Qi-Xing Huang   [1,3]Bart Adams   [1,2]Martin Wicke   [1]Leonidas J. Guibas

[1]Stanford University   [2]Max Planck Center for Visual Computing and Communication   [3]Katholieke Universiteit Leuven

Abstract

We present a robust and efficient algorithm for the pairwise non-rigid registration of partially overlapping 3D surfaces. Our approach treats non-rigid registration as an optimization problem and solves it by alternating between correspondence and deformation optimization. Assuming approximately isometric deformations, robust correspondences are generated using a pruning mechanism based on geodesic consistency. We iteratively learn an appropriate deformation discretization from the current set of correspondences and use it to update the correspondences in the next iteration. Our algorithm is able to register partially similar point clouds that undergo large deformations, in just a few seconds. We demonstrate the potential of our algorithm in various applications such as example based articulated segmentation, and shape interpolation.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

Surface registration is central to various aspects of computer graphics. Given a source and a target surface, the goal of registration is to find a motion that optimally positions points on the source surface into the scene in which the target surface lies. This is a common problem in 3D scanning [WJH*07], where multiple 3D scans of a moving object need to be registered, and in geometry processing, where correspondences between different surfaces are needed for applications, such as segmentation [SY07], information transfer [SP04] and morphing [KMP07].

So far, most of the surface registration algorithms have focused on rigid registrations, i.e., when the motion between the source and target is rigid. Dominant rigid registration algorithms are Iterative Closest Point (ICP) [BM92] and its variants [RL01]. ICP alternates between computing correspondences between the source and target and performing a rigid motion in response to these correspondences. If the distance between source and target is large, correspondences tend to be unstable and ICP easily gets trapped in local minima. To address this problem, another category of approaches [GMGP05, HFG*06] uses feature matches to generate initial guesses of the optimal motion which can be fed into ICP. Most challenges posed by rigid registration have been addressed, and there are several automatic 3D scanning systems [RHHL02, HH03].

Despite the amazing advances regarding rigid registration, the problem of automatic non-rigid registration has mainly only been tackled for small deformations [HTB03, BR07, MFO*07, WJH*07]. In this paper however, we consider au-

tomated registration under large deformations. One of the main challenges in this setting is obtaining stable correspondences. Incorrect correspondences easily trap the registration in local minima. In addition, appropriate deformation models are central to efficiency and robustness of non-rigid registration algorithms.

We present a novel framework for non-rigid registration, addressing the two challenges outlined above. We formulate non-rigid registration as an optimization problem and solve it by alternating correspondence computation and deformation optimization in terms of the resulting correspondences. Enforcing that correspondences preserve geodesic distances greatly improves stability. We also learn an appropriate deformation discretization during the optimization, leading to significantly better registration results. The major components of our algorithm are illustrated in Fig. 1.

Our algorithm has great advantages in various applications where correspondences across surfaces are required. We demonstrate its potential in example based articulated segmentation and shape interpolation. For example based segmentation, we do not need any correspondences between example poses, and it is even possible to handle incomplete examples. Our method can be used for shape interpolation because of its high suitability for dealing with extremely large deformations before steps.

In summary, our main contributions are

- An efficient and robust non-rigid registration framework for aligning partially overlapping point clouds. By enforcing that geodesic distances are preserved between sets of corresponding points, we are able to extract extremely sta-
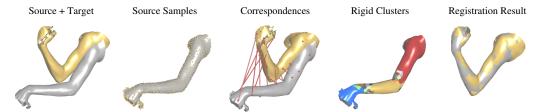
Source + Target	Source Samples	Correspondences	Rigid Clusters	Registration Result

Figure 1: *Illustration of the major components in our non-rigid registration algorithm. High quality correspondences are computed from a downsampled representation using feature guidance and spectral pruning. Rigid clusters are identified during registration. The source is iteratively deformed based on the computed correspondences and clusters to yield the final non-rigid registration.*

ble correspondences. This removes the need for manually established landmark correspondences.

- An adaptive deformation model that allows for stable deformations also for parts of the surface for which good correspondences are not available. Our deformation model does not have strong assumptions on rigidity or articulatedness, and can adapt to the deformation present in the model.
- We propose two algorithms that showcase the strength of our method. Building upon the registration framework, our methods for example based segmentation and deformation interpolation do not require recomputed correspondences, and work even with incomplete and noisy data.

1.1. Related Work

Correspondence Computation. The simplest correspondences are closest point correspondences [BM92, RL01] obtained by computing for each point on the source its closest point on the target surface. However, closest point correspondences tend to be unstable when the two surfaces are far apart. Therefore, most of the methods [HTB03, WJH*07, BR07] that use this kind of correspondences assume that the source and target surfaces are very close, an assumption we avoid in this paper.

In [SLW02], Sharp et al. propose to use a combination of Euclidean distances and feature distances to improve correspondence stability for rigid registration. However, in the presence of non-rigid motions, the correspondences computed with this method are not robust enough to ensure correct registration.

Using landmark correspondences can alleviate the stability problems [ACP03, PMG*05, BBBK08]. However, manually labeling and associating landmarks is a tedious and time consuming task. Geometric feature matching [GMGP05, HFG*06] has been successfully used to automatically generate landmark correspondences in the rigid case. In the non-rigid case, however, such landmarks are not necessarily persistent across frames, and geometric feature matching can be unreliable. To overcome this problem, we s enforce that the resulting correspondences are consistent and respect geodesic distances.

A very elegant technique for obtaining dense correspondences comes from computer vision. Optical flow methods can generate correspondences from video [ATD*08, ATR*08], rendered animations or densely sampled 3D scans [MFO*07]. However, these methods only work when the motions between frames are small.

Anguelov et al. [ASP*05] solved the problem of non-rigid correspondences by searching in the space of all possible correspondences using belief propagation. They also use geodesic distances as a quality criterion; however, their method assumes that the data surface can be fully embedded in the model surface.

Correspondences in 2D images have also been computed using modal analysis [SP95]. This method could also be applicable to 3D shapes, however, it requires complete shapes and a discretization suitable for finite element analysis.

Deformation Model. Deformation models have been studied in great detail in the context of shape deformation and we refer to [Sor06, NMK*05] for a good overview and to [HSL*06, BPGK06, SZT*07, SSP07, LCOGL07] for some recent advances.

In non-rigid registration, the deformation model used determines which assumptions about the deformations of the shape are made. Isometric models [PMG*05, BBBK08], as-rigid-as-possible models [ACP03, WJH*07] and skeleton driven models [EPT*07, PG08] have been used. In this paper, we allow for a wide variety of deformations by adapting the deformation model to the deformation observed during registration. In the presence of constraints obtained from correspondences, our model favors as-rigid-as-possible deformations, while in the absence of good correspondences, we assume an articulated motion, adaptively reducing the number of unknowns to obtain more robust results.

2. Overview

Throughout the paper, all vectors and matrices will be set in a bold font, while sets are set in script. The different steps in our algorithm are illustrated in Fig. 1 and outlined in more detail in Fig. 2.

Given source surface $\mathcal{S} = \{\mathbf{s}_i\}$ and a target surface $\mathcal{T} =$
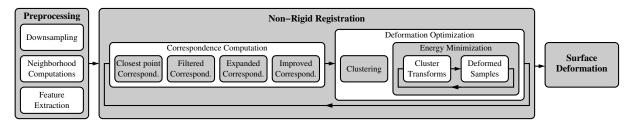
Figure 2: *Overview of the non-rigid registration pipeline. After downsampling the source and target point clouds and computing neighborhoods and features, we iteratively compute reliable correspondences and cluster and update the deformation field. After non-rigid registration we deform the original full resolution source point cloud to align with the target surface.*

$\{\mathbf{t}_i\}$, both represented as point clouds, the goal of non-rigid registration is to find correspondences that relate points on $\mathcal{S}$ to points on $\mathcal{T}$. Correspondences should associate measurements of the same physical point. These correspondences are then used to align the two surfaces by deforming $\mathcal{S}$ to $\mathcal{S}'$. If the correspondences are valid, this deformation results in close overlap of $\mathcal{S}'$ and $\mathcal{T}$. If the point cloud surfaces are incomplete and/or noisy, the alignment is partial and approximate.

We solve the non-rigid registration problem using an iterative two-step procedure. From the current deformed surface $\mathcal{S}'$, we compute reliable correspondences that relate sample points on $\mathcal{S}$ to sample points on $\mathcal{T}$. Given these correspondences, we update the deformed surface $\mathcal{S}'$ using energy minimization. This procedure is iterated until convergence. The result of our algorithm is a set of correspondences as well as a deformed source surface $\mathcal{S}'$ registered to the target surface $\mathcal{T}$.

Correspondence Computation. Correspondences $\mathcal{K} = \{(\mathbf{s}_i, \mathbf{t}_i)\}$ are computed for a subset of surface sample points $\mathbf{s}_i \in \hat{\mathcal{S}} \subset \mathcal{S}$ and $\mathbf{t}_i \in \hat{\mathcal{T}} \subset \mathcal{T}$. We discuss the sample selection in Sec. 3. For each sample point in the source and the target surface, we compute a feature vector $\mathbf{f}$ and match sample points that are close in Euclidean space *and* in signature space, i.e., when both $\|\mathbf{s}_i' - \mathbf{t}_i\|$ and $\|\mathbf{f}(\mathbf{s}_i) - \mathbf{f}(\mathbf{t}_i)\|$ are small. Here, $\mathbf{s}_i'$ corresponds to the current deformed position of the sample $\mathbf{s}_i$. For large deformations, feature matches generate much better correspondences than closest point matches. However, closest point matches yield good correspondences when the surfaces are close. As the registration progresses and $\mathcal{S}'$ approaches $\mathcal{T}$, our algorithm automatically reverts to closest point matching. Note that when computing $\mathbf{f}$, we only assume geometric information; no color or other texture information is used to compute correspondences. If this is desired, it is easy to incorporate additional information into the feature matching.

In the presence of non-rigid deformations, feature matches can be treacherous. Since the shape deforms, point signatures based on geometry change, and matches are unreliable. We assume that deformations are roughly isometric, and therefore, geodesic distances between points should be preserved by the correspondences. Therefore, we prune

the initial set of correspondences to enforce this property: If two sample points $\mathbf{s}_i$ and $\mathbf{s}_j$ in $\mathcal{S}$ have geodesic distance $d_g(\mathbf{s}_i, \mathbf{s}_j)$, then their corresponding target sample points $\mathbf{t}_i$ and $\mathbf{t}_j$ should have a similar geodesic distance on $\mathcal{T}$, i.e., $d_g(\mathbf{s}_i, \mathbf{s}_j) \approx d_g(\mathbf{t}_i, \mathbf{t}_j)$. We discuss the correspondence computation in more detail in Sec. 4.

Deformation Field Optimization. Given the correspondences, we deform the source surface to align it with the target surface. We use an energy minimization approach similar to [SSP07]. In regular intervals, source samples with similar transformations are grouped into clusters $\mathcal{C}_k$, for which we compute optimal rigid transformations. Neighboring cluster transformations are coupled to ensure a smooth deformation field. We then update the deformed source samples $\mathbf{s}_i'$ by minimizing the sum of a correspondence energy and a local rigidity preserving energy.

The clustering effectively constrains points within a cluster to move rigidly. In case of partial overlap between source and target scan, the clustering extends rigid transformations computed in regions with correspondences to regions where such correspondences are not available. Only relying on an energy term penalizing deformation can lead to drift, severely degrading the quality of the alignment. The clustering is performed adaptively and hence we allow both articulated motions as well as general deformable motions. We discuss the clustering and deformation optimization in more detail in Sec. 5.

3. Downsampled Representation

As both input point clouds potentially consist of millions of points, we perform the computations on subsets. We denote the chosen subsets of $\mathcal{S}$ and $\mathcal{T}$ as $\hat{\mathcal{S}} = \{\mathbf{s}_i\}$ and $\hat{\mathcal{T}} = \{\mathbf{t}_i\}$ respectively. The subsampling is performed using a combination of uniform sampling on the surface and uniform sampling in normal space as was proposed in [RL01]. The positional sampling is done by constructing an octree and picking one sample point per leaf cell, the one closest to the leaf's box center. The normal space sampling generates additional points by uniform sampling on the sphere obtained from the Gauss map of the input surface. These additional points, as noted also by [RL01], ensure proper sampling of all features. Surface normals are computed using the tech-

Figure 3: *Robust correspondence computation between source (silver) and target (gold). From left to right: Closest point correspondences, improved correspondences by local feature matching, filtered kernel correspondences by geodesic consistency and final propagated correspondences with the confidence weight color-coded from red (not confident at all) to blue (very confident).*

nique proposed in [MNG04]. Note that this subsampling can be just as easily applied to meshes or any other surface representation. Except for implementation details, the full surface registration pipeline is therefore equally applicable to those representations.

Neighborhoods. Our algorithm makes heavy use of geodesic distances on surfaces and sets of nearest neighbors with respect to this distance metric. In the following, we denote with $\mathcal{N}_g(\mathbf{p}, \mathcal{P})$ the set of k nearest neighbors to $\mathbf{p}$ that are part of the set of points $\mathcal{P}$.

These sets are computed by using a breadth first search on the Euclidean k-nearest neighbor graph of the set $\mathcal{P}$. We typically use $k = 15$. We precompute and store $\mathcal{N}_g(\mathbf{s}, \mathcal{S})$ and $\mathcal{N}_g(\mathbf{s}, \hat{\mathcal{S}})$ for all $\mathbf{s} \in \hat{\mathcal{S}}$, as well as $\mathcal{N}_g(\mathbf{t}, \mathcal{T})$ and $\mathcal{N}_g(\mathbf{t}, \hat{\mathcal{T}})$ for all $\mathbf{t} \in \hat{\mathcal{T}}$.

Geodesic distances are approximated using the graph distance $d_g(\mathbf{s}_i, \mathbf{s}_j)$ which corresponds to the length of the shortest path between $\mathbf{s}_i$ and $\mathbf{s}_j$ on the graph defined by $\mathcal{N}_g(\mathbf{s}, \hat{\mathcal{S}})$ for $\mathbf{s} \in \hat{\mathcal{S}}$.

Features. Using multi-level fitting of quadratic patches [CP03], we compute a feature vector

$$\mathbf{f}(\mathbf{s}_i) = [\kappa_1^N(\mathbf{s}_i), \kappa_2^1(\mathbf{s}_i), \dots, \kappa_1^N(\mathbf{s}_i), \kappa_2^N(\mathbf{s}_i)]^T \quad (1)$$

for each sample point in $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$. Here, $\kappa_{1,2}^j$ denote the principal curvatures for fitting level j, remapped to lie within the interval $[-1, 1]$. For each quadratic fit, we use all points that have a geodesic distance smaller than $\frac{d}{25-5j}$, where d is the diagonal length of the bounding box of the model. In our experiments, we use $N = 4$ fitting levels.

4. Correspondence Computation

Obtaining a good set of correspondences is crucial for faithful alignment of the source and target surface. To allow efficient computation for large point sets, we compute correspondences using the smaller subsets $\hat{\mathcal{S}} \subset \mathcal{S}$ and $\hat{\mathcal{T}} \subset \mathcal{T}$ defined above. The actual correspondence computation is performed in four major steps, illustrated in Fig. 3. First, we create a set of possible correspondences by connecting

points that are close in Euclidean space and feature space (see Sec. 4.1). Next, we prune this dense set by extracting a kernel of correspondences that are consistent in the sense that geodesic distances between source sample points are close to the geodesic distances between their corresponding target points. Sec. 4.2 discusses how this pruning can be efficiently done using a spectral method. Then, we expand the set of correspondences by propagating correspondence information from the kernel correspondences to other points on the surface (Sec. 4.3). Here, every correspondence is also given a confidence score. Finally, we fine-tune the correspondences by allowing correspondences to end in any point on the original target surface $\mathcal{T}$, as described in Sec. 4.4.

4.1. Initial Correspondence Computation

For each sample point $\mathbf{s}_i$, we define an initial correspondence by computing the closest point $\mathbf{t}_i$ on the target with respect to the current deformed position $\mathbf{s}_i'$.

We improve the initial correspondence $(\mathbf{s}_i, \mathbf{t}_i)$ by replacing $\mathbf{t}_i$ with the point $\mathbf{t}_j \in \mathcal{N}_g(\mathbf{t}_i, \hat{\mathcal{T}})$ in its neighborhood that minimizes $d_f(\mathbf{s}_i, \mathbf{t}_j) = \|\mathbf{f}(\mathbf{s}_i) - \mathbf{f}(\mathbf{t}_j)\|$. We iterate this until we find a local minimum of d_f.

Similarly, we update $\mathbf{s}_i$ by moving it to the source sample $\mathbf{s}_j$ that corresponds to the closest local minimum of $d_f(\mathbf{s}_j, \mathbf{t}_j)$. We discard the correspondence if $\|\mathbf{s}_i - \mathbf{s}_j\|$ is above a user-defined threshold. Large movement of $\mathbf{s}_i$ after $\mathbf{t}_i$ has been optimized indicates an unstable correspondence. The reasoning is similar to the heuristic used in [TL94]. An illustration of this procedure is shown in Fig. 3.

4.2. Correspondence Pruning

The initial set of correspondences might be inconsistent with the assumption that deformations are approximately isometric. Hence, we prune the set of correspondences to enforce that the geodesic distance between each pair of samples $\mathbf{s}_i$ and $\mathbf{s}_j$, should be similar to the distance between their corresponding target sample points, $d_g(\mathbf{s}_i, \mathbf{s}_j) \approx d_g(\mathbf{t}_i, \mathbf{t}_j)$.

By removing correspondences that are not consistent with our assumption, we can greatly improve their quality and robustness. We are looking for the largest set of correspondences that are consistent assuming isometric deformation. This kernel extraction can be performed using a standard spectral matching method [LH05]. We only briefly summarize the method here, please refer to [LH05] for more details.

Two correspondences $(\mathbf{s}_i, \mathbf{t}_i)$ and $(\mathbf{s}_j, \mathbf{t}_j)$ are consistent if the ratio of geodesic distances $d_g(\mathbf{s}_i, \mathbf{s}_j)/d_g(\mathbf{t}_i, \mathbf{t}_j)$ is close to 1. Hence, we can define the consistency measure between 0 and 1,

$$c_{ij} = \min\left\{ \frac{d_g(\mathbf{s}_i, \mathbf{s}_j)}{d_g(\mathbf{t}_i, \mathbf{t}_j)}, \frac{d_g(\mathbf{t}_i, \mathbf{t}_j)}{d_g(\mathbf{s}_i, \mathbf{s}_j)} \right\}, \quad (2)$$

where c_{ii} is defined as 1. Spectral matching proceeds by constructing a consistency matrix $\mathbf{M}$. To account for discretiza-

tion errors and to allow for small non-isometric deformations, we define **M** as

$$\mathbf{M}_{ij} = \begin{cases} (\frac{c_{ij} - c_0}{1 - c_0})^2 & c_{ij} > c_0, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where the threshold c_0 determines how much error in the consistency we are willing to accept. We use a value of 0.7. As shown in [LH05], the entries in the eigenvector **x** corresponding to the largest eigenvalue λ_{max} of **M** define a score for the initial correspondences.

Starting from an empty set, we iteratively add the correspondence that has the next highest score to our set of kernel correspondences $\mathcal{K}$, but only if its consistency measure (2) with respect to all kernel correspondences already in $\mathcal{K}$ is above the threshold c_0.

4.3. Correspondence Propagation

The kernel extraction results in a set of consistent correspondences. However, this set can be very sparse. Therefore, we expand the set of correspondences using a propagation strategy that respects the geodesic consistency criterion.

Given a sample $\mathbf{s}_i$ that does not have a corresponding point, we look for the nearest $\mathbf{s}_j$ with a correspondence $(\mathbf{s}_j, \mathbf{t}_j)$. We then assign a target sample that is most consistent with the kernel correspondences using the geodesic distance criterion

$$\mathbf{t}_i = \operatorname*{argmin}_{\mathbf{t} \in \mathcal{N}_g(\mathbf{t}_j, \hat{\mathcal{T}})} e_{\mathcal{K}}(\mathbf{s}_i, \mathbf{t}) \quad (4)$$

where the consistency error $e_{\mathcal{K}}$ is defined as

$$e_{\mathcal{K}}(\mathbf{s}, \mathbf{t}) = \sum_{(\mathbf{s}_k, \mathbf{t}_k) \in \mathcal{K}} [d_g(\mathbf{s}, \mathbf{s}_k) - d_g(\mathbf{t}, \mathbf{t}_k)]^2, \quad (5)$$

where the summation is over all kernel correspondences $(\mathbf{s}_k, \mathbf{t}_k)$. The correspondence propagation continues until all samples are assigned to a target point.

After propagation we assign a confidence weight to all correspondences. This weight is defined as a function of the correspondence's consistency compared to the consistency of the kernel correspondences

$$w_i = \exp(-\frac{e_{\mathcal{K}}(\mathbf{s}_i, \mathbf{t}_i)}{2e}), \quad e = \frac{1}{|\mathcal{K}|} \sum_{(\mathbf{s}_k, \mathbf{t}_k) \in \mathcal{K}} e_{\mathcal{K}}(\mathbf{s}_k, \mathbf{t}_k). \quad (6)$$

4.4. Correspondence Fine-Tuning

Due to efficiency reasons, the above computations were performed using the reduced sample sets $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$. This means that a correspondence always has its end point in the reduced set $\hat{\mathcal{T}}$. When $\mathcal{S}'$ is close to $\mathcal{T}$, the correspondences on $\hat{\mathcal{S}}$ and $\hat{\mathcal{T}}$ become inaccurate. Hence, correspondences are allowed to have end points in the full target point cloud $\mathcal{T}$ as opposed to the subset $\hat{\mathcal{T}}$.

For each correspondence $(\mathbf{s}_i, \mathbf{t}_i)$, we replace the end point $\mathbf{t}_i$ with $\mathbf{t}_j \in \mathcal{N}_g(\mathbf{t}_i, \mathcal{T})$, if $\mathbf{t}_j$ is the nearest neighbor of the deformed sample $\mathbf{s}'_i$ in $\mathcal{T}$.

Figure 4: *Correspondences between source (silver) and target (gold) for various examples. A small subset of the correspondences is shown for the sake of clarity.*

The final result of the correspondence computation is a set of correspondences $\{(\mathbf{s}_i, \mathbf{t}_i)\}$ that relate points $\mathbf{s}_i \in \hat{\mathcal{S}}$ to a target sample $\mathbf{t}_i \in \mathcal{T}$. Fig. 4 illustrates the correspondences for three different non-rigid registration examples. Note how reliable correspondences are found even under large deformations.

5. Deformation Optimization

Once correspondences are computed, we deform the surface to best align the corresponding points. In this section, we first describe how sample points can be grouped together in clusters whose deformation can be roughly described using a single rigid transformation. We then discuss how this clustered representation is used to find the optimal deformed sample positions using an energy minimization approach, after which we describe how to propagate the deformations onto the complete surface.

5.1. Cluster Computation

Clusters are groups of sample points whose deformation can be described by a single rigid transformation. For example, when aligning body scans, the lower and upper arm can each be represented by a single cluster, while the sample points near the elbow each define their own cluster. Grouping points and sharing their transformation greatly improves the stability of the optimization process when registering incomplete and noisy scans.

At the beginning of the clustering process, we create a single cluster $\mathcal{C}_i = \{\mathbf{s}_i\}$ for every sample $\mathbf{s}_i$. We then apply a forward search method [FCOS05] to iteratively combine neighboring clusters until a user-defined quality threshold is reached. In [FCOS05], forward search methods have been used in finding points that can be fitted well by planes. In our paper, we use forward search methods to find correspondences that can be described by rigid motions.

For every cluster $\mathcal{C}_k$, we define the extended cluster $\tilde{\mathcal{C}}_k$ by adding all neighboring points to the extended cluster:

$$\tilde{\mathcal{C}}_k = \mathcal{C}_k \bigcup_{\mathbf{s}_i \in \mathcal{C}_k} \mathcal{N}_g(\mathbf{s}_i, \hat{\mathcal{S}}). \quad (7)$$

We will call two clusters neighboring if their extended clusters intersect.

Each cluster $\mathcal{C}_k$ is assigned a rigid transformation that

Figure 5: *Two views of the clusters color coded on the source surface for the registration example on the left. The registration deforms the source by an almost 180 degree rotation as indicated by the arrows. Many clusters are used in the regions on the lower back and around the arm pit where the source surface bends to align with the target. Large clusters can be used in regions that roughly deform rigidly.*

maps its undeformed sample points $\mathbf{s}_i$ to their deformed sample positions $\mathbf{s}'_i$. We obtain translation $\mathbf{T}_k$ and rotation $\mathbf{R}_k$ by minimizing

$$E_k = \sum_{\mathbf{s}_i \in \tilde{\mathcal{C}}_k} \|\mathbf{R}_k\mathbf{s}_i + \mathbf{T}_k - \mathbf{s}'_i\|^2. \qquad (8)$$

This is a simple rigid registration problem with known correspondences, which can be solved efficiently [Hor87].

For cluster merging, we sort the clusters in order of increasing rigid registration error E_k. We pick the first cluster $\mathcal{C}_k$ and compute the maximal residual registration error E_{kl} for all its neighboring clusters $\mathcal{C}_l$:

$$E_{kl} = \max_{\mathbf{s}_i \in \tilde{\mathcal{C}}_l} \|\mathbf{R}_k\mathbf{s}_i + \mathbf{T}_k - \mathbf{s}'_i\|^2. \qquad (9)$$

We merge $\mathcal{C}_k$ with all neighboring clusters that have a maximal rigid registration error E_{kl} below a threshold E_{max}. We then recompute E_k, resort the list of clusters, and iterate.

The result of our clustering algorithm is a set of K clusters $\mathcal{C}_k$, where every sample belongs to exactly one cluster. In the subsequent deformation optimization process we will use the extended overlapping clusters $\tilde{\mathcal{C}}_k$ to enforce continuity in the rigid transformations.

In the first 10 iterations of the deformation optimization we do not invoke the clustering algorithm, i. e., we use a single cluster $\mathcal{C}_i = \{\mathbf{s}_i\}$ per sample point. Otherwise the clustering would result in one big cluster because in the beginning $\mathbf{s}'_i = \mathbf{s}_i$. Starting from the 10th iteration, we cluster at regular intervals of 10 iterations.

Fig. 5 Illustrates the clustering for a non-rigid alignment example involving partial scans and large deformations.

5.2. Energy Minimization

After computing new correspondences, we update the deformed sample points $\mathbf{s}'_i$ so that they are best aligned with their corresponding target points $\mathbf{t}_i$. We use a combination of point-to-point and point-to-plane constraints, motivated from the local quadratic approximation of the squared distance function [MGPG04].

Hence, we wish to find the deformed sample positions $\mathbf{s}'_i$ that minimize

$$E_{\text{corr}} = \sum_{\mathbf{s}_i \in \hat{\mathcal{S}}} w_i \left[\alpha\|\mathbf{s}'_i - \mathbf{t}_i\|^2 + \beta((\mathbf{s}'_i - \mathbf{t}_i)^T\mathbf{n}_i)^2 \right], \qquad (10)$$

where w_i is the consistency weight (Eq. 6), $\mathbf{n}_i$ is the surface normal corresponding to the target point $\mathbf{t}_i$ and α and β are parameters that vary the contribution of the point-to-point and point-to-plane energies. We use $\alpha = 0.6$ and $\beta = 0.4$.

To avoid the trivial solution $\mathbf{s}'_i = \mathbf{t}_i$, we add an additional energy that penalizes local non-rigid deformations. Using the cluster energy E_k, we define the total rigidity error as

$$E_{\text{rigid}} = \sum_k E_k = \sum_k \sum_{\mathbf{s}_i \in \tilde{\mathcal{C}}_k} \|\mathbf{R}_k\mathbf{s}_i + \mathbf{T}_k - \mathbf{s}'_i\|^2. \qquad (11)$$

Note that because the extended neighboring clusters overlap, the resulting transformations vary smoothly.

The total deformation energy is now given by

$$E = \lambda_{\text{corr}}E_{\text{corr}} + \lambda_{\text{rigid}}E_{\text{rigid}}, \qquad (12)$$

where λ_{corr} and λ_{rigid} vary the contribution of both energy terms. We typically use $\lambda_{\text{corr}} = 2$ and $\lambda_{\text{rigid}} = 1$.

The goal is to find the deformed sample positions $\mathbf{s}'_i$ that minimize E. Due to the dependence of the cluster rotations $\mathbf{R}_k$ on $\mathbf{s}'_i$, this is a non-linear problem. We solve for the deformed sample positions by fixing $\mathbf{R}_k$ and $\mathbf{T}_k$ in each step and solving for $\mathbf{s}'_i$, and then computing new optimal rotations $\mathbf{R}_k$ and translations $\mathbf{T}_k$ [SA07, XZY*07]. On average, only 6 iterations were required for convergence.

5.3. Surface Deformation

The result of the registration algorithm is a set of deformed positions $\mathbf{s}'_i$ for the samples $\mathbf{s}_i$ in the downsampled source surface $\hat{\mathcal{S}}$. After the registration process is complete, we extrapolate these deformations to the entire source point cloud $\mathcal{S}$. We use a simple partition of unity approach [SSP07]. The new position of $\mathbf{s}_j \in \mathcal{S}$ is given by

$$\mathbf{s}'_j = \sum_{\mathbf{s}_k \in \mathcal{N}_g(\mathbf{s}_j, \mathcal{S})} \overline{\theta}_{jk}(\mathbf{R}_k\mathbf{s}_j + \mathbf{T}_k), \qquad (13)$$

where $\mathbf{R}_k$ and $\mathbf{T}_k$ are the rotation and translation associated with $\mathbf{s}_k$'s cluster.

The normalized weights $\overline{\theta}_{jk} = \theta_{jk}/\sum_l \theta_{jl}$ with $\theta_{jk} = \exp(-\|\mathbf{s}_j - \mathbf{s}_k\|^2/\sigma^2)$ are defined to smoothly decay with increasing distance. σ is a user-defined parameter that determines the support of the sample points.

6. Results and Discussion

We tested our algorithm on various data sets with both partial similarity and full similarity. For each model, we are given a set of poses, and we compute registrations for each possible pair of poses. Table 1 summarizes the statistics. All timings were measured on a 3.2GHz PC with 2GB of RAM.

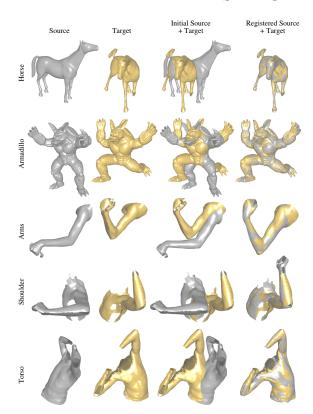Some example pairs are shown in Fig. 6. As can be seen

Source	Target	Initial Source + Target	Registered Source + Target

Figure 7: *Comparison of different strategies for the non-rigid alignment of the Torso data set. (a) Source (gray) and target (gold). Each image (b)-(d) shows the deformed source surface and the target surface. (b) Using closest point correspondences. (c) Using our strategy, but without clustering. (d) Using both our feature based correspondences and the rigid clustering results in much better results for this difficult non-rigid registration problem.*

scans of a moving person, and no ground truth was available. We therefore validated the registration visually.

Note that some of the models are partially and approximately symmetric. For such models, there are local minima that can trap our algorithm. We found that similar to rigid registration, these local minima can be avoided by trying all initial rotations from the icosahedral rotational symmetry group. In practice, we only try more than one initial rotation if the total registration error is very high. Since our registration method is very fast, this is not a problem. Using this approach we obtained valid registrations for all pairs of the Arms, Shoulders and Torso data sets.

Since we use subsampled versions of the point clouds to compute the registrations, the full point clouds are not precisely aligned. A local deformation using closest point correspondences can be used to optimize the final alignment. On the other hand, subsampling makes the alignment more robust in the presence of noise.

Fig. 7 shows a comparison with two alternatives for the registration of the Torso data set. Using simple closest point correspondences instead of the consistent geodesic and feature based correspondences leads to severe misalignments. If we filter the correspondences for geodesic consistency, but do not use clustering during the deformation optimization, shrinkage occurs in underconstrained regions, e. g., near the wrist. Using a single rigid transformation for large clusters of sample points reduces the effect of bad or sparse correspondences and results in better registration quality.

One of the limitations of our method is that if the assumption of geodesic consistency between the surfaces is invalid due to topological changes, our method will result in erroneous registration. Changes in topology cause abrupt changes in geodesic distances on the surface, leading to very few, or even wrong correspondences after filtering. In the future, we plan to address these problems by only considering

Figure 6: *Non-rigid registration under large deformation for various data sets.*

from these figures, our algorithm is able to align incomplete surfaces that undergo significant deformations.

For the Horse and Armadillo, we use the original undeformed point cloud as the source surface and various deformed instances as the target surface. Note that we did not exploit the fact that the source and target shapes share a common discretization. All correspondences were computed using our algorithm. However, using this ground truth, we were able to validate our registration algorithm and found that the resulting deformed shapes were all correctly aligned.

The Arms, Shoulders and Torso data sets are incomplete

data set	#poses	#pairs	$|\mathcal{S}|$	$|\hat{\mathcal{S}}|$	pre time	reg time
Horse	10	45	80k	2500	7.4s	13.6s
Armadillo	12	66	332k	2500	7.6s	14.8s
Arms	36	630	80k	600	2.1s	1.1s
Shoulder	33	528	117k	800	3.4s	1.9s
Torso	27	231	325k	1100	4.5s	4.5s

Table 1: *Data set statistics. Shown are the number of poses and the resulting number of registration pairs for each model, the average number of points per pose (the numbers vary between poses for real-world scans), the cardinality of the sets $\mathcal{S}$ and $\hat{\mathcal{S}}$, as well as the average timing per pair, for preprocessing (computation of neighborhoods, geodesic distances and features) and registration.*
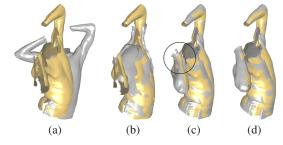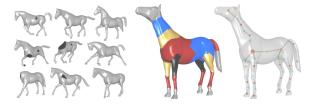
Figure 8: *Example based articulated segmentation. Left: Possibly incomplete example poses. Middle: Rigid clusters on the source shape. Right: Extracted skeleton.*

the most consistent samples both in kernel extraction and correspondence propagation.

7. Applications

Since the proposed algorithm reliably and automatically registers shapes that undergo large deformations, it is applicable to a wide variety of problems in computer graphics. To demonstrate the potential of the algorithm, we have implemented an example-based mesh segmentation method that can work with incomplete examples, as well as a deformation interpolation method.

7.1. Example Based Articulated Segmentation

Our framework can be used to compute the rigid parts of an articulated object from multiple example poses. The resulting segmentation can then be used to extract a skeleton which in turn can be used to create new articulated poses. Our proposed segmentation algorithm is more general and flexible than previous approaches [JT05, SY07] in the sense that we do not require known correspondences between the source and the target poses, nor do we require the example shapes to be complete.

In order to perform segmentation by example, we consider one source surface $\mathcal{S}$, but now register with respect to L target surfaces $\mathcal{T}_l$. Similar to the original algorithms, we cluster the source sample points in K clusters $\mathcal{C}_k$, but we now construct the clusters such that each can be rigidly aligned with *all* the target poses. Hence, the alignment energy (Eq. 8) of a cluster $\mathcal{C}_k$ now considers all target surfaces

$$E_k = \frac{1}{L} \sum_{l=1}^{L} \sum_{\mathbf{s}_i \in \tilde{\mathcal{C}}_k} \|\mathbf{R}_k^l \mathbf{s}_i + \mathbf{T}_k^l - \mathbf{s}_i^l\|^2, \qquad (14)$$

where $\mathbf{R}_k^l$ and $\mathbf{T}_k^l$ denote the rotation and translation for cluster $\mathcal{C}_k$ with respect to the target surface $\mathcal{T}_l$ and $\mathbf{s}_i^l$ is the deformed position of sample $\mathbf{s}_i$ for the alignment with $\mathcal{T}_l$.

The clustering can result in over-segmentation for example in the region of joints. As a post-processing step we reduce the number of clusters by only considering those clusters with at least 40 sample points. The remaining sample points are then added to neighboring clusters by iteratively growing the surviving clusters. The resulting rigid clusters can then be used to extract a skeleton.

Figure 9: *Interpolated shapes (silver) between source and target (gold).*

Fig. 8 illustrates the segmentation algorithm for the Horse model. From a set of 9 incomplete example poses we extract clusters and a skeleton. The skeleton nicely captures the rigid structure of the Horse model as learned from the examples.

7.2. Shape Interpolation

Given a source surface $\mathcal{S}$ and a target surface $\mathcal{T}$, we can apply our approach to generate a series of surfaces $\mathcal{M}_l, 0 \leq l \leq L$, where $\mathcal{M}_0 = \mathcal{S}$ and $\mathcal{M}_L = \mathcal{T}$, which smoothly interpolate between $\mathcal{S}$ and $\mathcal{T}$.

We start by performing a non-rigid alignment between $\mathcal{S}$ and $\mathcal{T}$, assuming that the deformation is roughly isometric such that our method is able to find reliable correspondences. Next, we generate intermediate points $\mathbf{s}_i^l$ for all the sample points $\mathbf{s}_i$ obtained during registration by minimizing

$$E_I = \sum_{l=0}^{L-1} \left[\sum_{k} \sum_{\mathbf{s}_i \in \mathcal{C}_k} \|\mathbf{R}_k \mathbf{s}_i^l + \mathbf{T}_k - \mathbf{s}_i^{l+1}\|^2 \right. \\ \left. + \gamma \sum_{\mathbf{s}_i \in \hat{\mathcal{S}}} \|\mathbf{s}_i^l - \mathbf{s}_i^{l+1}\|^2 \right], \qquad (15)$$

subject to the hard constraints $\mathbf{s}_i^0 = \mathbf{s}_i$ and $\mathbf{s}_i^L = \mathbf{s}_i'$. E_I is a variant of the energy used in [KMP07].

The first term constrains the interpolation to be as rigid as possible. Note that if the motion between $\mathcal{S}$ and $\mathcal{T}$ is articulated, then this information will be reflected in the clustering, and the resulting interpolation will also be articulated. The second term forces the interpolation path to be as short as possible. We set $\gamma = 0.001$ as suggested in [KMP07].

Fig. 9 illustrates the shape interpolation for the Armadillo model. Note that a high quality interpolation is obtained even though the deformation from the source to the target shape is large.

8. Conclusion

We have presented a novel algorithm for non-rigid registration of a pair of partially overlapping surfaces. Enforcing consistency with respect to geodesic distances on the surface greatly improves the quality of feature-based correspondences. Especially when considering large deformation, the new algorithm is significantly more robust than previous approaches. It allows for fully automatic alignment even for hard cases involving partial overlap and noise.

As future work we plan to apply our method to non-rigid

scan alignment and surface reconstruction involving multiple point clouds [WJH*07]. Special care will be required to avoid drift.

Acknowledgements

References

[ACP03] ALLEN B., CURLESS B., POPOVIĆ Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *Proc. SIGGRAPH* (2003), pp. 587–594.

[ASP*05] ANGUELOV D., SRINIVASAN P., PANG H.-C., KOLLER D., THRUN S., DAVIS J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS.*, vol. 17. 2005, pp. 33–40.

[ATD*08] AHMED N., THEOBALT C., DOBREV P., SEIDEL H.-P., THRUN S.: Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *Proc. CVPR* (2008). To appear.

[ATR*08] AHMED N., THEOBALT C., ROESSL C., THRUN S., SEIDEL H.-P.: Dense correspondence finding for parametrization-free animation reconstruction from video. In *Proc. CVPR* (2008). To appear.

[BBBK08] BRONSTEIN A. M., BRONSTEIN M. M., BRUCKSTEIN A. M., KIMMEL R.: Analysis of two-dimensional non-rigid shapes. *IJCV.* (2008). To appear.

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *PAMI 14*, 2 (1992), 239–256.

[BPGK06] BOTSCH M., PAULY M., GROSS M., KOBBELT L.: Primo: coupled prisms for intuitive surface modeling. In *Proc. SGP* (2006), pp. 11–20.

[BR07] BROWN B. J., RUSINKIEWICZ S.: Global non-rigid alignment of 3-d scans. *TOG. 26*, 3 (2007), 21.

[CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. SGP* (2003), pp. 177–187.

[EPT*07] ECKSTEIN I., PONS J.-P., TONG Y., KUO C. C. J., DESBRUN M.: Generalized surface flows for mesh processing. In *Proc. SGP* (2007), pp. 183–192.

[FCOS05] FLEISHMAN S., COHEN-OR D., SILVA C. T.: Robust moving least-squares fitting with sharp features. *TOG. 24*, 3 (2005), 544–552.

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Proc. SGP* (2005), pp. 197–206.

[HFG*06] HUANG Q.-X., FLÖRY S., GELFAND N., HOFER M., POTTMANN H.: Reassembling fractured objects by geometric matching. *TOG 25*, 3 (2006), 569–578.

[HH03] HUBER D., HEBERT M.: Fully automatic registration of multiple 3d data sets. *IVC 21*, 7 (July 2003), 637–650.

[Hor87] HORN B.: Closed form solutions of absolute orientation using unit quaternions. *J. Opt. Soc. of A. A 4*, 4 (1987), 629–642.

[HSL*06] HUANG J., SHI X., LIU X., ZHOU K., WEI L.-Y., TENG S.-H., BAO H., GUO B., SHUM H.-Y.: Subspace gradient domain mesh deformation. *TOG 25*, 3 (2006), 1126–1134.

[HTB03] HÄHNEL D., THRUN S., BURGARD W.: An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *IJCAI* (Acapulco, Mexico, 2003).

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. In *Proc.SIGGRAPH* (2005), pp. 399–407.

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. In *Proc. SIGGRAPH* (2007), vol. 26, pp. #64, 1–8.

[LCOGL07] LIPMAN Y., COHEN-OR D., GAL R., LEVIN D.: Volume and shape preservation via moving frame manipulation. *TOG 26*, 1 (2007).

[LH05] LEORDEANU M., HEBERT M.: A spectral technique for correspondence problems using pairwise constraints. In *ICCV* (2005), pp. 1482 – 1489.

[MFO*07] MITRA N. J., FLORY S., OVSJANIKOV M., GELFAND N., GUIBAS L., POTTMANN H.: Dynamic geometry registration. In *Proc. SGP* (2007), pp. 173–182.

[MGPG04] MITRA N. J., GELFAND N., POTTMANN H., GUIBAS L. J.: Registration of point cloud data from a geometric optimization perspective. In *Proc. SGP* (2004), pp. 23–32.

[MNG04] MITRA N. J., NGUYEN A., GUIBAS L.: Estimating surface normals in noisy point cloud data. In *IJCGA.* (2004), vol. 14, pp. 261–276.

[NMK*05] NEALEN A., MÜLLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically Based Deformable Models in Computer Graphics. In *Eurographics: State of the Art Report* (2005).

[PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum 27*, 2 (2008), 243–253.

[PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M. H., GUIBAS L. J.: Example-based 3d scan completion. In *Proc. SGP* (2005), pp. 23–32.

[RHHL02] RUSINKIEWICZ S., HALL-HOLT O., LEVOY M.: Real-time 3d model acquisition. *TOG. 21*, 3 (2002), 438–446.

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling* (2001), pp. 145–152.

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proc. SGP* (2007), pp. 109–116.

[SLW02] SHARP G. C., LEE S. W., WEHE D. K.: Icp registration using invariant features. *PAMI. 24*, 1 (2002), 90–102.

[Sor06] SORKINE O.: Differential representations for mesh processing. *Computer Graphics Forum 25*, 4 (2006), 789–807.

[SP95] SCLAROFF S., PENTLAND A.: Modal matching for correspondence and recognition. *PAMI 17*, 6 (1995), 545–561.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *Proc. SIGGRAPH* (2004), pp. 399–405.

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. In *Proc. SIGGRAPH* (2007), p. 80.

[SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. In *Proc. SGP* (2007), pp. 153–162.

[SZT*07] SHI X., ZHOU K., TONG Y., DESBRUN M., BAO H., GUO B.: Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics. In *Proc. SIGGRAPH* (2007), p. 81.

[TL94] TURK G., LEVOY M.: Zippered polygon meshes from range images. In *Proceedings of ACM SIGGRAPH 94* (1994), pp. 311–318.

[WJH*07] WAND M., JENKE P., HUANG Q., BOKELOH M., GUIBAS L. J., SCHILLING A.: Reconstruction of deforming geometry from time-varying point clouds. In *SGP.* (2007), pp. 49–58.

[XZY*07] XU W., ZHOU K., YU Y., TAN Q., PENG Q., GUO B.: Gradient domain editing of deforming mesh sequences. *TOG. 26*, 3 (2007), 84.