Towards Unsupervised Segmentation of Semi-Rigid Low-Resolution Molecular Surfaces

Leonidas J. Guibas* Yusu Wang†

Abstract

In this paper, we study a particular type of surface segmentation problem motivated by molecular biology applications. In particular, two input surfaces are given, coarsely modeling two different conformations of a molecule undergoing a semi-rigid deformation. The molecule consists of two subunits that move in a roughly rigid manner. The goal is to segment the input surfaces into these semi-rigid subcomponents. The problem is closely related to non-rigid surface registration problems, although considering only a special type of deformation that exists commonly in macromolecular movements (such as the popular *hinge motion*). We present and implement an efficient paradigm for this problem, which combines several existing and new ideas. We demonstrate the performance of our new algorithm by some preliminary experimental results in segmenting low-resolution molecular surfaces.

1 Introduction

Registration of shapes is an important problem arising in many research areas, such as computer graphics, vision, pattern recognition, and structural biology. In general, given two structures represented as, say, surfaces, one wishes to identify for every point from one surface the *corresponding point* from the other. This registration problem is closely related to the problem of measuring shape similarities.

Much of previous work has been focused on the so-called *rigid-body registration*, where given two structures A and B, the goal is to find the best rigid transformation for B so that the distance between A and B is minimized (and thus the similarity between A and B is maximized). In this paper, we consider the case where the input object consists of a small number of components. Each component moves roughly in a rigid manner. Given two conformations of this object, represented by surfaces S_A and S_B, we wish to segment S_A (and/or S_B) into these components without any prior knowledge on their correspondences, and call the resulting problem the *semi-rigid segmentation* problem. Note that the solution of this problem, namely the segmentation as well as the correspondences between the respective subunits, can then be used to find a full registration between the two input surfaces.

Motivation. The semi-rigid segmentation problem has many applications. For example, in human body tracking, the major body parts undergo hinge-type motions. Our main motivation, however, comes from molecular structural biology. In particular, a macro-molecule (e.g, a protein) may

*Department of Computer Science, Stanford University, Stanford, CA 94305. E-mail: guibas@graphics.stanford.edu.

†Department of Computer Science and Engineering, the Ohio State University, Columbus, OH 43210. E-mail: yusu@cse.ohio-state.edu.

change its conformation significantly during important biological processes. With current structure determination technologies, it is possible to obtain some "snapshots" of this deformation process, such as the beginning and ending conformations. In order to understand the entire deformation, it is necessary to find correspondences among these few obtained conformations. On the other hand, high-resolution structures, which include the types and positions of atoms in a molecule, are very hard and time-consuming to obtain. More and more research turn to low-resolution structural data, like cryo-EM data, where the atomic information of the molecule is not available. As such, effectively, we are only given two surfaces in arbitrary orientation with no correspondences information, and we wish to register them. One major type of macro-molecular deformation is the so-called "hinge-motion", where components of the molecule, usually protein domains, rotate around some "hinge pivot". The registration between conformations under hinge-type motion is essentially a semi-rigid segmentation problem.

Related work. Registration has been widely studied and used in a broad range of applications, where input objects can be image data, curves, surfaces, and so on. We refer the readers to [15, 42] for some surveys on this topic. Below we give a brief review for surface registration methods, with a focus on non-rigid cases, as well as related work in the field of molecular structural biology.

One basic challenge in the registration problem is that there are two subproblems needed to be optimized simultaneously, and the solutions of them interact with each other. In particular, one needs to find both the right *transformation* to align the two surfaces, and the *correspondences* between points from input surfaces. To make things worse, the input data are usually noisy and possibly incomplete (thus may require partial matching). Hence the problem is complex even for rigid-body registrations. To make the problem computationally more manageable, many approaches first extract *features* from input objects and check only those transformations that align compatible features. In other words, they only sample the transformational space at a few potential positions. The problem then becomes how to capture representative and discriminative features [8, 6, 14, 18, 21, 24]. There are also several popular paradigms for the rigid-body registration that can be combined with the use of features, such as the Iterative closest point (ICP) algorithm [9] and Geometric hashing technique [40]. Nevertheless, despite a great amount of research devoted to the rigid-body registration problem, the field still contains many unanswered questions.

The non-rigid registration problem is much harder. Much research on this topic is motivated by applications from computer vision to track patterns in video sequences [25] or in medical image processing for describing deformations of organs such as the brain and heart [30, 31]. Different methods have been developed for different types of objects. Given the temporal coherence that exists in video sequences, the deformation between two consecutive conformations is usually not too great. Some (physics-based) deformable model is usually built for the input object to help to establish correspondences between two consecutive conformations, as well as the deformation between them [42, 25, 31, 32]. Most works assume either (some of) the correspondences are known, or that the objects are already roughly aligned. For example, a few correspondences can be provided a priori, obtained either manually [27, 28] or by attaching markers to the object [5, 16, 20]. There is also a lot of research on the articulated motion, such as tracking human bodies or hand gestures [19, 43], which is similar to the type of semi-rigid deformations we consider. In general, the problem of non-rigid registration remains largely open.

Registration also plays an important role in computational biology. The widely investigated problem of protein structure classification is essentially curve matching, where proteins are represented by their backbone curves [17, 33, 29]; and surface registration is crucial to the study of protein-protein recognition, which can be modeled as a partial surface matching problem under the constraint that the two molecular surfaces do not penetrate each other much [11, 35, 39]. Most pre-

vious work has focused on rigid-body registration. For non-rigid cases, the normal modes analysis (NMA), including the so called elastic network theory, has been one of the main tools for modeling deformations (other than molecular dynamics). NMA is a powerful tool and has obtained success in several cases [1, 4, 36, 37]. Such approaches usually start with a high-resolution structure where both the type and the position of each atom is known, although recently there are some work on constructing elastic network by discretizing a low-resolution model [12, 26, 38]. The deformable model (i.e, the normal modes) are constructed based purely on one single conformation. Hence it does not take advantage of the other conformations available, neither can it produce the registration between two conformations directly. Recently, in [36], in order to fit a high-resolution X-ray structure into a low-resolution cryo-EM map (which can be considered as the registration between the two structures), they first represent the X-ray structure in a virtual coordinate system using the low-order normal modes as bases. They then optimize the structure in this coordinate system to best match the EM map (roughly a density map). Initial alignments between the two input structures are required, and the involved optimization problem is time-consuming to solve (the time is usually measured in hours). Finally, for hinge-type motion, there is research aiming at segmenting the input structure into semi-rigid subunits, as well as identifying the hinge pivot, when both the high-resolution structures and correspondences between input conformations are given [41, 22].

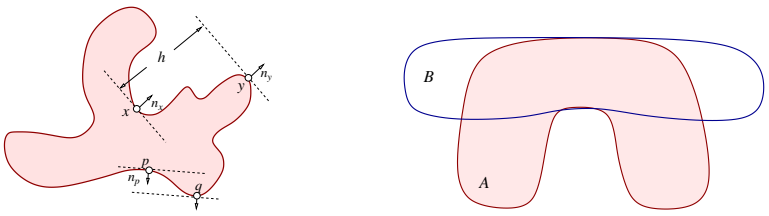
Our contribution. We focus on a specific type of deformation of molecular structures, where several subunits (domains) of a molecule deform in a roughly rigid manner. This includes the most popular types of macromolecular movements, such as the hinge motion, as well as many instances of the so-called "shear" motion. Our goal is to develop a simple geometric approach that can identify large deforming subunits reliably and automatically. Compared to previous more general approaches, our method is efficient and robust, and works when no high-resolution structures and prior correspondences are available at all. The resulting segmentation of input structure into semi-rigid components can then serve as inputs for more refined registration procedures, or for visualization tools so that biologists can inspect them visually to obtain insights. Furthermore, since the registration between two surfaces becomes relatively easy once the segmentation and correspondences are determined, our approach can facilitate the search of a given structure (say, some protein) in the low-resolution structure data of a large and complex system (such as a ribosome), in which case efficiency is a crucial factor.

More specifically, we consider the aforementioned semi-rigid segmentation problem for two given molecular surfaces S_A and S_B. We design and develop a segmentation framework that combines several ideas, some existed already, in a novel way. The new approach uses landmark-based virtual coordinates instead of the usual Cartesian coordinates to handle deformations. In order to identify the landmarks automatically, we exploit a voting idea, taking advantages of a set of potentially good rigid registrations. In particular, by exploiting the feature pairs computed from the so-called elevation function [2], we develop a scheme to vote for *landmarks* using *landmark-based* coordinates. This scheme is facilitated by the one-to-one correspondence between rigid transformations and pairs of feature pairs existed in our framework. Finally, the extracted landmarks also give rise to a natural procedure to segment input molecular surfaces. The entire framework is easy to implement and we present preliminary experimental results at the end to demonstrate its performances.

2 Coarse Rigid Registrations

First, we compute a set of coarse rigid registrations between S_A and S_B. Our approach is based on the *elevation function* $\mathrm{Elev} : S \to \mathbb{R}$ over a surface S introduced in [2]. Roughly speaking, every

point $x \in S$ has a canonical pairing partner y that shares the same normal direction n_x with x: the pair (x, y) describes a feature in direction n_x, and $\mathrm{Elev}(x)$, defined as the height difference between x and y in this direction, indicates the size of this feature (See Figure 1 (a) for an illustration in the plane). Furthermore, the set of maxima of the elevation function, together with their canonical pairing partners, form a set of *feature pairs*, capturing important protrusions and cavities from the given surface. Each feature pair $s = (x, y)$ consists of a pair of points x and y, together with their common normal and the elevation value. We sometimes refer to s as a *segment* and x, y as its endpoints.



(a) (b)

Figure 1: (a) In 2D: x is paired with y with common normal. $\mathrm{Elev}(x) = \mathrm{Elev}(y) = h$. The maxima of elevation return a set of feature pairs like (x, y) and (p, q). (b) The top-scored registration between A and B may not align any pair of corresponding components.

Pairs of feature-pairs and rigid transformations. In three dimensions, a *pair of feature-pairs (*PFP*)* (s_1, s_2), with s_1 from S_A and s_2 from S_B, is sufficient to determine a rigid transformation $\mu(s_1, s_2)$. This can be achieved by aligning the corresponding endpoints of s_1 and s_2, as well as the normal directions. If the normal of s_1 (resp. of s_2) is in the same direction of segment s_1 (resp. s_2), we consider this PFP to be degenerate and do not consider it. Given surfaces S_A and S_B, we first compute a set $\mathcal{F}(A)$ (resp. $\mathcal{F}(B)$) of feature pairs from S_A (resp. S_B) using the maxima of elevation function. The set of points involved in feature pairs from $\mathcal{F}(A)$ and $\mathcal{F}(B)$ are denoted by $\mathcal{P}(A)$ and $\mathcal{P}(B)$, respectively. We then consider only transformations produced by aligning a pair of feature-pairs, one from $\mathcal{F}(A)$ and one from $\mathcal{F}(B)$. The resulting set of transformations is denoted by Π. For each rigid transformation $\mu \in \Pi$, we compute its *score* based on some scoring function $\sigma(S_A, \mu(S_B))$ to measure how good μ is. We sort Π in decreasing order of their scores. At this state, we simply take $\sigma(S_A, \mu(S_B))$ as the number of pairs of points $p \in \mathcal{P}(A)$ and $q \in \mu(\mathcal{P}(B))$ such that $d(p, q)$, the Euclidean distance between p and q, is smaller than some given threshold. We use the geometric hashing technique to compute Π and their scores. In particular, geometric hashing is a technique for computing the scores of alignments between two sets of points (possibly augmented with other information). Instead of checking every pair of feature-pairs (one from $\mathcal{F}(A)$ and one from $\mathcal{F}(B)$) and computing the score of resulting alignment, geometric hashing technique processes $\mathcal{F}(A)$ and $\mathcal{F}(B)$ separately in two stages, evaluating the score of each possible alignment implicitly. This leads to a significant reduction in time complexity, and is a widely used technique in computer vision, graphics, as well as molecular structure biology [40, 13].

Let $n = |S_A|$ be the number of vertices in surface S_A; and $m = |S_B|$. The size of $\mathcal{F}(A)$ and $\mathcal{F}(B)$, thus $\mathcal{P}(A)$ and $\mathcal{P}(B)$, are upper bounded by $O(n)$ and $O(m)$ respectively. In practice, they are much smaller: around 100 in our experiments, as we only keep those pairs with large elevation value (thus more significant features). The size of Π is bounded by $|\mathcal{F}(A)| \cdot |\mathcal{F}(B)| = O(nm)$.

Finally, we remark that in this work, all transformations we consider are produced by aligning PFPs. Hence we sometimes do not distinguish between a transformation and a PFP. For example, the set of transformations Π obtained above can also be viewed as a set of PFPs.

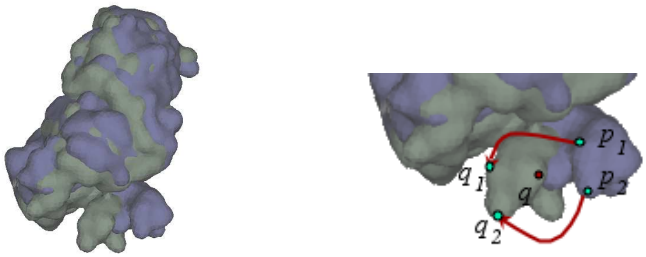
3 Segmentation Algorithm

The method described above performs well for rigid transformations [39]. However, in our case, there exists certain non-rigid deformation between input surfaces. Hence a rigid transformation is unable to identify all matching components at once. In order to segment the input surface, say S_A, into semi-rigid components, a natural approach consists of the following steps: (i) start with a coarse registration μ_1 that align one pair of components well, (ii) eliminate all points of S_A that are close to their correspondences on S_B under transformation μ_1, (iii) extract a good registration from the remaining points to identify the second pair of matching components, and (iv) repeat step (ii) and (iii) to identify more semi-rigid components.

There are, however, a few problems with this approach: (P1.) How to choose the first registration μ_1? The top-ranked transformation from Π is a natural choice, but it may in fact align none of the components well (see Figure 1 (b)). (P2.) How to define *correspondences* between points from S_A and S_B w.r.t. μ_1? If we use Euclidean distance to measure closeness, a point on a deformed component may find a completely wrong correspondence, which will induce wrong registration at step (iii). (P3.) How to extract the next matching components? (P4.) How to eventually segment S_A into semi-rigid components?

Below we first address problem P2, after which we describe our new algorithm.

3.1 Landmark-based coordinates



(a) (b)

Figure 2: (a) Two surfaces aligned by their main component. Under this registration, in (b), points p_1 and p_2 from S_A should correspond to q_1 and q_2 from S_B respectively. But they will most likely be matched to q under Euclidean distances.

For rigid alignments, one common way to define the correspondence of $p \in S_A$ is by finding its nearest neighbor in S_B under Euclidean distance metric. This approach unfortunately fails for the non-rigid scenario. For example, in Figure 2 we are given two surfaces where their main components are well-aligned, but not the small components. As such, for a point p in this small component, its nearest neighbor under the Euclidean distance can be quite far from its real correspondence q (Figure 2 (b)). This remains to be the case even if we augment the Euclidean distance with normal information. To find more reliable correspondences, we would like to use a different distance measure that is hopefully invariant w.r.t. the type of semi-rigid deformation we consider. Intuitively, given a few points $p_1, ..., p_k$ from S_A, as S_A undergoes hinge-type motion , the geodesic distance from any point $p \in S_A$ to p_i's does not vary significantly. This suggests the following landmark-based virtual coordinates for points from each surface.

Given surfaces S_A, a set of *landmarks* of S_A is simply a set of points $\mathrm{L} = \{\mathrm{l}_1, \ldots, \mathrm{l}_k\}$, $\mathrm{l}_i \in S_A$. Let $\mathrm{gd}(p, q)$ denote the geodesic distance from $p \in S_A$ to $q \in S_A$, which is the length of the

shortest path connecting p to q along surface S_A. We define the *landmark-based coordinate* of $p = (p_x, p_y, p_z)$ w.r.t. L as the following (k+3)-tuple

$$\mathbf{p} = \xi_{\mathrm{L}}(p) = \langle p_x, p_y, p_z, \mathrm{gd}(p, \mathrm{l}_1), \ldots, \mathrm{gd}(p, \mathrm{l}_k) \rangle.$$

The distance between two points p and q represented in landmark-based coordinates (i.e, for $\mathbf{p}$ and $\mathbf{q}$) is defined as

$$\delta_L(p, q) = \sqrt{\lambda \sum_{i=1}^{3} (\mathbf{p}[i] - \mathbf{q}[i])^2 + (1 - \lambda) \sum_{i=4}^{k+3} (\mathbf{p}[i] - \mathbf{q}[i])^2}, \quad (1)$$

where λ specifies the weight of the Euclidean distance between p and q. We sometimes omit the subscript L from $\xi_{\mathrm{L}}()$ and $\delta_{\mathrm{L}}()$ when its choice is clear from the context.

Given surfaces S_A and S_B, a set of landmarks $\mathrm{L} = \{\mathrm{l}_1, \ldots, \mathrm{l}_k\}$ for S_A corresponding to a set of landmarks $\mathrm{M} = \{\mathrm{m}_1, \ldots, \mathrm{m}_k\}$ for S_B means that l_i corresponds to m_i for $1 \leq i \leq k$. We sometimes refer to set L and M as *matching sets of landmarks* for S_A and S_B; $\delta_L(p, q)$ can be extended for $p \in S_A$ and $q \in S_B$ once L and M are given.

3.2 Overview of the algorithm

Ideally, to find the right correspondences for all points from S_A, we wish to have at least one landmark from each semi-rigid component of S_A, as well as one from its corresponding component from S_B. Our algorithm aims at identifying each pair of matching components with one PFP, (s_1, s_2), with $s_1 \in \mathcal{F}(A)$ and $s_2 \in \mathcal{F}(B)$. The overview of our segmentation algorithm is shown in Figure 3. PREPROCESSING and STEP 1 were described in previous section. Next we explain steps 2 – 4 in detail. We remark how the four aforementioned problems (P1 to P4) are addressed in this new approach at the end of this section.

SegRigidComponents (S_A, S_B)

PREPROCESSING) Compute sets of feature pairs $\mathcal{F}(A)$ and $\mathcal{F}(B)$ for S_A and S_B
STEP 1) Construct and sort Π by aligning PFPs from $\mathcal{F}(A)$ and $\mathcal{F}(B)$
STEP 2) Compute a set of *reliable* PFPs
STEP 3) Compute matching sets of landmarks L and M
STEP 4) Segment S_A based on L and M

Figure 3: Overview of our algorithm for the semi-rigid segmentation problem.

3.3 Landmark Selection

The goal here is to compute the matching sets of landmarks L and M for S_A and S_B, respectively. Note that a pair of matching landmarks, $(\mathrm{l}_i, \mathrm{m}_i)$, is nothing but two points corresponding to each other reliably. A natural approach is to first identify some feature points from each surface, and then establish reliable correspondences among them based on some shape descriptor around each point [14]. However, molecular surfaces are quite homogeneous in the sense that many points look alike in their local neighborhood. In order to be more discriminative, it is then desirable to use more complex features, such as pairs of points, or triple-points, instead of simply points. Unfortunately, increasing the complexity of the definition of features also increases the number of potential features: for example, there are $O(n^3)$ triple-points for a surface with n vertices.

In our approach, we choose to use the feature-pairs ($\mathcal{F}(A)$ and $\mathcal{F}(B)$) computed from the elevation function as our basic features. Note that although the number of pairs of points is $\Theta(n^2)$, there are only linear ($O(n)$) number of feature pairs. More specifically, in STEP 2, we first compute a set Ω of *reliable* PFPs, where a *reliable* PFP means that the two feature-pairs involved in it potentially correspond to each other. We then select from Ω a small number of *consistent* ones as the final sets of matching landmarks (STEP 3). The details are described below.

Voting for reliable PFPs**.** First, we describe how to compute Ω. Recall that we have sorted the set of registrations between S_A and S_B, Π, by their scores. Consider the top N transformations from Π, $\Pi(N)$. They roughly provide a set of "good" registrations between S_A and S_B. Intuitively, if a feature-pair $s_1 \in \mathcal{F}(A)$ corresponds to $s_2 \in \mathcal{F}(B)$, then s_1 should be close to s_2 w.r.t many *good* registrations from $\Pi(N)$. Our algorithm is based on this idea.

First, to measure the distance between two feature-pairs s_1 and s_2, we use landmark-based distance based on some *temporary* landmarks. In particular, recall that any transformation $\mu \in \Pi(N)$ was generated by, thus associated with, a pairs of feature-pairs, say $(a_0, a_1) \in \mathcal{F}(A)$ and $(b_0, b_1) \in \mathcal{F}(B)$. We use a_i's and b_i's, $1 \leq i \leq 2$, as temporary landmarks for S_A and S_B respectively, under current registration μ. That is, each point in $p \in S_A$ now has the following 5-tuple as its coordinate: $\mathbf{p} = \langle p_x, p_y, p_z, \mathrm{gd}(p, a_0), \mathrm{gd}(p, a_1)\rangle$, and similarly for a point $q \in S_B$. For a PFP (s_1, s_2) where $s_1 = (p_1, p_2)$ and $s_2 = (q_1, q_2)$, the *distance* between s_1 and s_2 is defined as $\delta(s_1, s_2) = \delta(p_1, \mu(q_1)) + \delta(p_2, \mu(q_2))$, where $\delta()$ is the landmark-based distance as introduced earlier in Eqn (1). Furthermore, we say that s_1 and s_2 are compatible if (1) difference between the length of p_1p_2 and q_1q_2 is small and (2) difference between $\mathrm{Elev}(p_1)$ and $\mathrm{Elev}(q_1)$ is also small. Now, to obtain reliable PFPs, the algorithm maintains a *vote counter* for every PFP between $\mathcal{F}(A)$ and $\mathcal{F}(B)$. In particular, for any transformation in $\Pi(N)$, the algorithm increases the vote counter of a PFP (s_1, s_2) by one if s_1 and s_2 are compatible, and if the distance between them, $\delta(s_1, s_2)$ is smaller than some threshold λ_1. The algorithm then scans through all transformations from $\Pi(N)$, and ranks the set of PFPs by their vote counts at the end. Obviously, higher votes indicates more reliable PFPs. In our implementation, we compute the votes for PFPs by standard hashing idea, where different from the traditional geometric hashing algorithms which vote for transformations, our approach is a somewhat dual version where we use a set of transformations to vote for feature pairs. The high level framework is outlined in Figure 4.

Finally, we take Ω as the set of PFPs with a vote greater than some threshold λ_2. We next aim at selecting from Ω one reliable PFP from each semi-rigid component as landmarks. So if there are k semi-rigid components, ideally, we choose k feature pairs; denote by LP the resulting set of k PFPs.

Selecting landmarks from Ω**.** To find LP[0], the PFP to identify the first semi-rigid component, we scan through Ω in order of decreasing votes, and return the first PFP whose corresponding transformation produces a registration with score greater than some threshold λ_3. In our experiments, the first pair from Ω is usually returned as LP[0].

Suppose we have one reliable PFP, say (s_1, s_2), that does not lie on the component identified by LP[0]. Intuitively, it should be *consistent* w.r.t. LP[0], namely, the landmark-based distance between s_1 and $\mu(s_2)$ is small (i.e, smaller than some threshold λ_4), where $\mu = \mu[0]$; while at the same time, the Euclidean distances between corresponding points from s_1 and $\mu(s_2)$ are large (i.e, greater than some threshold λ_5). Hence our algorithm first eliminates from Ω those PFPs that are *not* consistent with LP[0]. It then choose from the remaining PFPs the first one (thus with highest vote) that has a large euclidean distance between corresponding points w.r.t. $\mu[0]$, and set it as LP[1].

VoteReliablePFPs (Π, N, $\mathcal{F}(A)$, $\mathcal{F}(B)$)
  // stage 1: Registration:
For every $\mu \in \Pi(N)$ **Do**
  For every $s = (p, q)$ in $\mathcal{F}(A)$ **Do**
    construct landmark-based coordinates $\mathbf{p}, \mathbf{q}$ w.r.t. μ
    compute an index based on $\mathbf{p}, \mathbf{q}$
    register s in a hash table based on its index
  Endfor
Endfor
  // stage 2: Voting:
For every $\mu \in \Pi(N)$ **Do**
  apply μ to $\mathcal{F}(B)$
  For every $s = (p, q)$ in $\mathcal{F}(B)$ **Do**
    construct landmark-based coordinates $\mathbf{p}, \mathbf{q}$ w.r.t. μ
    compute an index based on $\mathbf{p}, \mathbf{q}$
    access corresponding hash table bin associated with this index
    For every feature-pair $s' \in \mathcal{F}(A)$ in this bin **Do**
      increase votes for the PFP (s', s) if they are compatible
    Endfor
  Endfor
  rank all PFPs by their votes, return Ω: those with vote $\geq \lambda_2$
Endfor

Figure 4: Compute a set of reliable PFPs Ω.

One can then repeat this procedure to identify more components till no PFPs from Ω are left. Once LP is computed, we collect the set of endpoints from s_1's (resp. s_2's) for all $(s_1, s_2) \in \mathrm{LP}$ as the landmark set L for S_A (resp. M for S_B).

3.4 Segmentation

Given LP, L and M as constructed above, let $k = |\mathrm{LP}|$; $|\mathrm{L}| = |\mathrm{M}| = 2k$. We compute the landmark coordinates (a $(2k + 3)$-tuple) for every point on S_A and S_B. To segment the input surface, say S_A, into the k corresponding components, we construct a function $f_i : S_A \to \mathbb{R}$, for each $1 \leq i \leq k$, as follows. Align S_A and S_B based on $\mu[i]$, the transformation corresponding to $\mathrm{LP}[i]$. For every point $p \in S_A$, find its nearest neighbor $\mathrm{NN}(p)$ from S_B based on the $(2k+3)$-tuple landmark coordinates. Set $f_i(p) = \delta(p, \mathrm{NN}(p))$. We say that a point p belongs to component-i if $f_i(p) = \min_{j=1}^{k} f_j(p)$. To obtain the i'th segment, we simply collect all points from component-i.

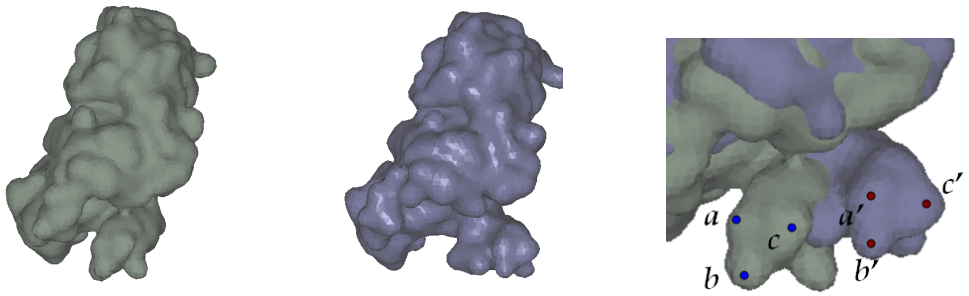
Remarks. We now come back to the four problems raised at the beginning of this section. Our algorithm uses the top N transformations from Π to vote for reliable PFPs, thus it does not relies on any single rigid transformation (P1). We use landmark-based distance instead of euclidean distance, which is potentially more stable under semi-rigid deformations (P2). In particular, the set of feature-pairs obtained from the elevation function produces a meaningful set of point-pairs, and they provide landmarks both in the voting process and in the final segmentation. Finally, landmark-based distance also helps us to distinguish PFPs from different components (P3), as well as induce the

final segmentation (P4).

4 Experimental Results

Time complexity. Given two input surfaces S_A and S_B, of n and m vertices respectively, the most time consuming step is the PREPROCESSING step: compute the set of feature pairs of S_A and S_B using the elevation function [2]. The worst case complexity for computing elevation maxima is $O(n^4)$, although in practice, it is much faster. We consider this step as the preprocessing step, as once the set of feature pairs of a surface is computed, one can use it for multiple registration tasks.

It is hard to give an exact time complexity for the remaining algorithm, as geometric hashing technique is involved, and it also depends on choices of different parameters. More specifically, STEP 1 computes the set of coarse rigid transformations using geometric hashing technique with running time $O(t_1(|\mathcal{F}(A)|^2 + r_1|\mathcal{F}(B)|^2))$, where t_1 is the time to access a particular entry in the hash table given an index, and r_1 is the maximum size of a bin associated with any index in the hash table; both t_1 and r_1 are usually considered to be constants. In STEP 2, we approximate geodesic distance by the graph distance (i.e., using only edges from the input mesh). Thus it takes $O(N(n+m)\log nm)$ to approximate geodesic distance from all vertices to all points contributing to the top N transformations of Π using Dijkstra's algorithm. The voting algorithm runs in roughly $O(N(|\mathcal{F}(A)| + |\mathcal{F}(B)|))$ time. A straightforward implementation of STEP 3 takes $O(|\Omega|^2 + |\mathcal{F}(A)||\mathcal{F}(B)|)$ time, and STEP 4 runs in $O(nm)$ time (for every $p \in S_A$, we find its nearest neighbor under Landmark-based distance in $O(m)$ time). In practice $N, |\mathcal{F}(A)|, |\mathcal{F}(B)|$, and $|\Omega|$ are usually around 100.



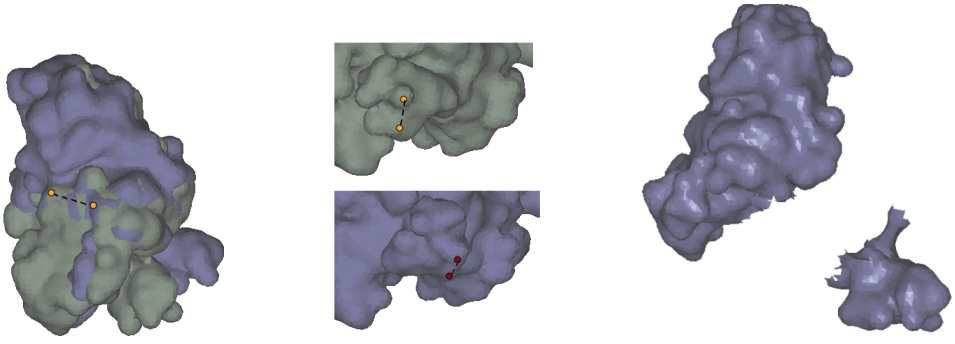
(a) (b) (c)

Figure 5: (a) and (b) show the two input surfaces (isosurfaces from the pseudo-EM-map computed for molecules with PDB codes 1FMW and 1VOW respectively. In (c), we see that the points from the small leg are locally similar, thus hard to decide how to align abc with $a'b'c'$ just by local information.

Inputs setup. Our targeted application is to segment low resolution data, such as cryo-EM data, in order to facilitate the analysis of their deformation when high resolution atomic structure of an input molecule is not available. In this section, we use the so-called "pseudo-maps" as in [10] to test our algorithm so that the correct answer is known. In particular, we take a pair of different conformations of the same molecule that undergoes some large hinge-type deformations. For each conformation, to obtain its pseudo-EM-map, we take its high-resolution structure and use the EMAN software package [23] to introduce certain amount of Gaussian noise to its original density map. We then compute the iso-surface w.r.t some prefixed value in the coarsened density map (pseudo-EM-map) to produce input surface for our algorithm.

We applied our algorithm to three sets of test data. The first set includes the prehydrolyzing state (PDB code 1FMW, [7]) and the ATP hydrolyzing state (PDB code 1VOM [34]) of myosin. The data we use are after pre-procession at Pande's group at Stanford University for normal modes analysis applications, where some extra residues are removed so that the two conformations have the same length. The second set is obtained from the Database of Macromolecular Movements [4]. They are two conformations of DNA Polymerase beta (PDB codes 1BPD and 2BPG respectively). As we will see later, these two sets present different properties in their motion. The third set is also from the Database of Macromolecular Movements. They are the two conformations of ATP sulfurylase (PDB codes 1I2D and 1M8P respectively). For all three sets of data, we read in their PDB file, generated their pseudo-EM-map using EMAN software [23], and extract two sets of surfaces. The input surfaces are in random relative orientations. We then compute the set of feature pairs for all input surfaces using algorithm from [39].

Segmentation results. The Myosin data set has a relatively small deformation among the two input sets. Roughly speaking, this molecular motor has a main body and two small legs, one of them moves outwards during the deformation (Figure 5 (a) and (b)). Although the moving component and the deformation is relative small, we note that the small leg is rather homogeneous. It does not have any locally very distinguishable features to help to establish correspondences for this small leg. Thus locally there are ambiguity how to align triangle abc from S_A to $a'b'c'$ in S_B. This problem is alleviated in our approach by also considering their geodesics to landmarks on the main components (thus incorporating more global information). The two input surfaces, S_A and S_B, have 8314 and 8226 vertices respectively. After preprocessing, we have $|\mathcal{F}(A)| = 132$ and $|\mathcal{F}(B)| = 72$, thus representing S_A and S_B in a much more concise way. Our algorithm identifies two components for this data sets, and the corresponding landmarks are shown in Figure 6 (a) and (b). The resulting two components produced are shown in 6 (c). The computation of feature pairs from the Elevation function takes roughly 100 minutes for each input surface. Once this pre-processing is done, the rest of the algorithm finishes within seconds, much more efficient than the optimization approach in [36] that takes at least several hours. Its fast speed enables users to play with different parameters and compute more than one possible segmentations as seeds for later more refined registration or other types of processes.



(a) (b) (c)

Figure 6: In (a), we align the two surfaces by their first pair of feature pairs computed by our algorithm (identifying the main component); one of them is roughly marked. In (b), we mark the pair of corresponding feature pairs identifying the smaller components from the two surfaces respectively. The resulting segmentation is shown in (c).

The DNA Polymerase beta data presents large scaled deformation in its two conformations. The motion includes hinge-type bending as well as some twisting. Furthermore, the topology of the two input surfaces are also different: the genus of the ending surface is one (see Figure 7 (a) and (b)). So there can be reasonable amount of distortion in geodesics as well. Nevertheless, our algorithm is able to identify two components as shown in Figure 7 (c), demonstrating its robustness. The third data set also presents both hinge-type bending and twisting. The result is shown in Figure 7 (d) - (f). The two segmented components correspond roughly to the two main domains in the molecule. The running time for these two data sets is of the same order as the first one.

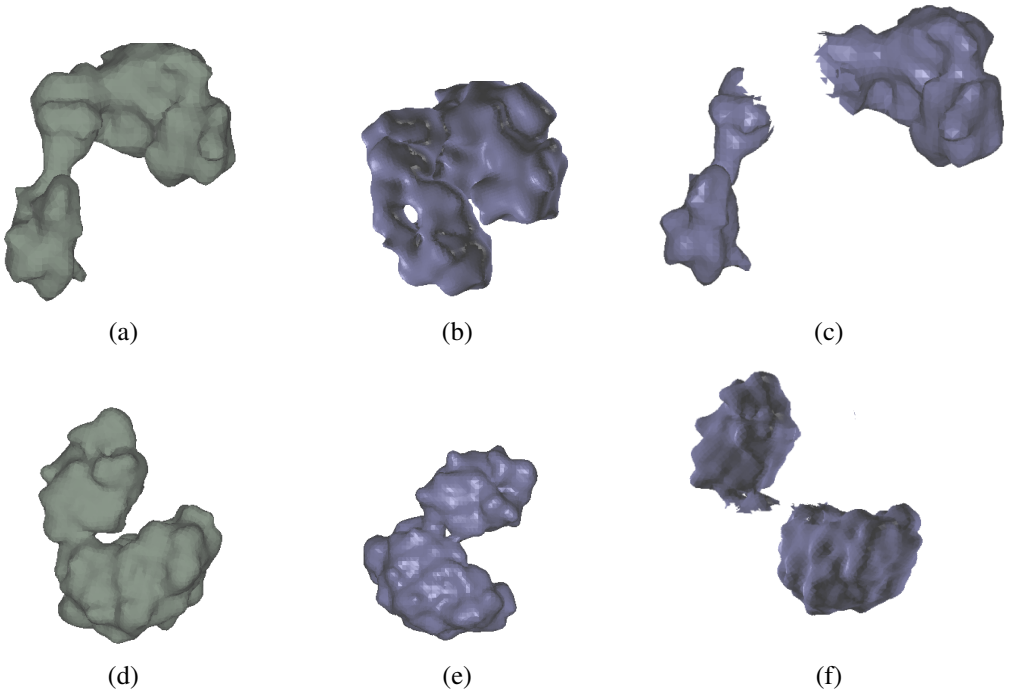


Figure 7: (a) and (b) show the input surfaces extracted from the pseudo-EM-map for molecules with PDB codes 1BPD and 2BPG respectively. The resulting segmentation is shown in (c). (d) and (e) are molecules with PDB codes 1I2D and 1M8P, and (f) shows the resulting segmentation.

Robustness and parameters. We note that for efficiency reason, our algorithm approximates the geodesic distances by graph distances in input meshes. It is well known that for "bad" meshes, such as those with long and skinny triangles, the graph distance can be arbitrarily far from the real geodesic distance. This distortion also depends on the resolution of input surfaces, where a finer triangulation tends to lead to lower distortion. Hence in practice, it is desirable that the two input triangular meshes should have roughly uniform sampling, good triangles, and similar resolution. In our experiments, input surfaces are obtained as iso-surfaces from some density map of the same resolution. Hence the resulting meshes are usually of compatible meshing qualities. For other inputs, some pre-processing, such as refining input meshes, may be necessary. We remark that the algorithm is reasonably robust with respect to the resolution of input meshes, as long as both meshes have similar quality. We have conducted experiments on the third data sets on meshes of two different resolutions, and obtained similar segmentation results. However, if input meshes are too coarse, then the algorithm fails to capture good feature pairs, and geodesic distances also becomes too coarse to

be useful. In general, it is important to note that the quality of input meshes strongly influence the performance of our segmentation algorithm.

There are several parameters involved in our algorithm. Ideally, we hope that the values of these parameters can be decided automatically. Currently, however, the user needs to input these thresholds. Part of the reason is because that these parameters are case-dependent. For example, geodesic distances are much less preserved in the second test data compared to the other two sets. To increase tolerance in the geodesic distance, it is necessary to increase the threshold related to geodesic distances. Since the deformation is also large in this case, we are still able to identify semi-rigid subcomponents reliably. We have also conducted several experiments for the third data sets using different sets of parameters around the one used for the reported result in Figure 7 (f). Similar segmentation results are obtained, although the landmarks computed may vary.

5 Discussion

We have proposed in this paper a new method to extract a few reliable landmarks for surfaces undergoing hinge-type deformation, which further help define a segmentation of the input surface into semi-rigid subunits. Our preliminary experimental results show that the algorithm is efficient and effective, identifying semi-rigid subunits automatically, once input parameters are given.

Although our targeted application is to segment low-resolution molecular data to facilitate analysis of their deformation when no atomic structure is available, the proposed method can be applied to other fields where semi-rigid deformations are involved. The landmarks computed by our algorithm are of independent interest, and can be inputs for other segmentation or tracking algorithms. For example, these landmarks can help to produce a deformation between two conformations using approaches such as 'as-rigid-as possible' shape interpolation [3].

Our current experiments with the pseudo-EM data serves as a proof-of-principle test for our new algorithm. In the next step, we plan to apply the algorithm to real EM data to provide an efficient tool to, for example, facilitate biologists detecting the presence of a certain structure (e.g, a particular protein) in the low resolution structure of a complex system (e.g, in the ribosome), as well as the deformation involved. One main challenge involved is the reliability of geodesic distances for isosurfaces extracted from real EM data, which can be very noisy: for example, the surfaces may connect to each other by small bridges, creating short cuts and changing geodesics greatly. It is an interesting and important problem to characterize major types of topological features created this way, and to develop methods to remove them.

Finally, currently, our algorithm focuses on a very specific type of deformation where geodesic distances are relatively well-preserved. We will investigate other types of invariants (other than geodesics) which can embrace more general types of non-rigid deformations. We leave this as one important future direction. Note that both the landmark based distance and the voting scheme in our framework are general, and can be modified to accommodate other types of invariants. We also point out that currently, our method segments input molecular surfaces at a single level, while methods such as normal modes analysis can describe motions at multiple scales. It will be interesting to see whether our framework can be further improved to capture deformation at more levels.

Acknowledgment. The work was supported by NSF grants FRG 0354543 and ITR 0205671, and NIH grant GM072970. The authors would like to thank Vijay Pande for motivating the study of the semi-rigid segmentation problem, Natasha Gelfand and Niloy Mitra for helpful discussions, and the anonymous reviewers for useful comments.

References

[1] H. K. T. A, M. J. Field, and D. Perahia. Tertiary and quaternary conformational changes in aspartate transcarbamylase: a normal mode study. *Proteins*, 34:96–112, 1999.

[2] P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang. Extreme elevation on a 2-manifold. In *Proc. 20th Sympos. Comput. Geom.*, pages 357–365, 2004.

[3] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[4] V. Alexandrov, U. Lehnert, N. Echols, D. Milburn, D. Engelman, and M. Gerstein. Normal modes for predicting protein motions: a comprehensive database assesement and associated web tool. *Protein Science*, 14(3):633–643, 2005.

[5] B. Allen, B. Curless, and Z. Popović. The space of human body shapes. *ACM Transactions on Graphics*, 22(3):587–594, 2003.

[6] G. Barequet and M. Sharir. Partial surface and volume matching in three dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):929–948, 1997.

[7] C. B. Bauer, H. W. Holden, J. B. Thoden, R. Smith, and I. Rayment. X-ray structures of the apo and MgATP-bound states of *dictyostelium discoideum* myosin motor domain. *J. Biol. Chem.*, 275:38494–38499, 2000.

[8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis an Machine Intelligence*, 24(4):509–522, 2002.

[9] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[10] H. Ceulemans and R. B. Russell. Fast fitting of atomic structures to low-resolution electron density maps by surface overlap maximization. *Journal of Molecular Biology*, 338:783–793, 2004.

[11] R. Chen, L. Li, and Z. Weng. ZDOCK: An initial-stage protein docking algorithm. *Proteins*, 52(1):80–87, 2003.

[12] M. Delarue and P. Dumas. On the use of low-frequency normal modes to enforce collective movements in refining macromolecular structural models. *Proceedings of National Academy of Science*, 101(18):6957–6962, 2004.

[13] D. Fischer, S. L. Lin, H. Wolfson, and R. Nussinov. A geometry-based suite of molecular docking processes. *J. Mol. Biol.*, 248:459–477, 1995.

[14] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proc. Symp. Geom. Processing*, pages 197–206, 2005.

[15] B. Girod, G. Greiner, and H. Niemann, editors. *Principles of 3D image analysis and synthesis*. Kluwer Academic Publishers, 2000.

[16] B. Guenter, C. Grimm, D. Wood, H. Wmlvar, and F. Pighin. Making faces. In *SIGGRAPH '98: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 55–66, 1998.

[17] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, 233:123–138, 1993.

[18] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[19] I. A. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: a physics-based approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 980–984, 1994.

[20] G. A. Kalberer and L. V. Gool. Face animation based on observed 3d speech dynamics. In *IEEE Conference on Computer Animation*, pages 20–27, 2001.

[21] J. J. Koenderink. *Solid shape*. MIT Press, Cambridge, MA, USA, 1990.

[22] W. G. Krebs and M. Gerstein. The morph server: a standardized system for analyzing and visualizing macromolecular motions in a database framework. *Nucleic Acids Research*, 28(8):1665–1675, 2000.

[23] S. Ludtke, W. Jiang, L. Peng, G. Tang, P. Baldwin, S. Fang, H. Khant, and L. Nason. EMAN software package. http://ncmi.bcm.tmc.edu/homes/stevel/EMAN/, 2006.

[24] S. Manay, B. Hong, A. J. Yezzi, and S. Soatto. Integral invariant signatures. In *European Conference on Computer Vision*, pages 87–99, 2004.

[25] D. N. Metaxas. *Physics-Based Deformable Models*. Kluwer Academic, 1997.

[26] D. Ming, Y. Kong, S. J. Wakil, J. Brink, and J. Ma. Domain movements in human fatty acid synthase by quantized elastic deformational model. *Proceedins of National Academy of Science*, 99(12):7895–7899, 2002.

[27] J.-Y. Noh and U. Neumann. Expression cloning. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 277–288, 2001.

[28] M. Pauly, N. J. Mitra, J. Giesen, M. Gross, and L. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005.

[29] F. M. G. Pearl, D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo. Assigning genomic sequences to CATH. *Nucleic Acids Research*, 28(1):277 – 282, 2000.

[30] D. Ruechert and D. J. Hawkes. Registration of biomedical images. In R. Baldock and J. Graham, editors, *Image Processing and Analysis - A Practical Approach*. Oxford University Press, 1999.

[31] D. Rueckert. Non-rigid registration: Techniques and applications. In J. V. Hajnal, D. L. G. Hill, and D. J. Hawkes, editors, *Medical Image Registration*. CRC Press, 2001.

[32] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(6):545–561, 1995.

[33] I. N. Shindyalov and P. E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of optimal path. *Protein Engineering*, 11(9):739–747, 1998.

[34] C. A. Smith and I. Rayment. X-ray structure of the magnesium(II).ADP.vanadate complex of the *dictyostelium discoideum* myosin motor domain to 1.9Å resolution. *Biochemistry*, 35:5405–5417, 1996.

[35] G. R. Smith and M. J. E. Sternberg. Prediction of protein-protein interactions by docking methods. *Current Opinion in Structural Biology*, 12:29–35, 2002.

[36] F. Tama, O. Miyashita, and C. L. B. III. Normal mode based flexible fitting of high-resolution structure into low-resolution experimental data from cryo-em. *Journal of Structural Biology*, 147:315–326, 2004.

[37] F. Tama and Y. H. Sanejouand. Conformational change of proteins arising from normal mode calculations. *Protein Engineering*, 14:1–6, 2001.

[38] F. Tama, W. Wriggers, and C. L. B. III. Exploring global distortions of biological macromolecules and assemblies from low-resolution structural information and elastic network theory. *Journal of Molecular Biology*, 321:297–305, 2002.

[39] Y. Wang, P. K. Agarwal, P. Brown, H. Edelsbrunner, and J. Rudolph. Coarse and reliable geometric alignment for protein docking. In *Pac Symp Biocomput*, pages 66–77, 2005.

[40] H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4(4):10 – 21, 1997.

[41] W. Wriggers and K. Schulten. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. *Proteins*, 29:1–14, 1997.

[42] T. Yoo, editor. *Insight into images: Principles and practices for segmentation, registration, and image analysis*. A. K. Peters, 2004.

[43] L. Zelnik-Manor, M. Machline, and M. Irani. Multi-body factorization with uncertainty: Revisiting motion consistency. To appear in IJCV special issue on Vision and Modeling of Dynamic Scenes, 2006.