

Graph-based Deformable Matching of 3D Line Segments with Application in Protein Fitting

Hang Dou · Matthew L Baker · Tao Ju

Abstract We present an algorithm for matching two sets of line segments in 3D that have undergone non-rigid deformations. This problem is motivated by a biology application that seeks a correspondence between the alpha-helices from two proteins, so that matching helices have similar lengths and can be aligned by some low-distortion deformation. While matching between two feature sets have been extensively studied, particularly for point features, matching line segments has received little attention so far. As typical in point-matching methods, we formulate a graph matching problem and solve it using continuous relaxation. We make two technical contributions. First, we propose a graph construction for undirected line segments such that the optimal matching between two graphs represents an as-rigid-as-possible deformation between the two sets of segments. Second, we propose a novel heuristic for discretizing the continuous solution in graph matching. Our heuristic can be applied to matching problems (such as ours) that are not amenable to certain heuristics, and it produces better solutions than those applicable heuristics. Our method is compared with a state-of-art method motivated by the same biology application and demonstrates improved accuracy.

Keywords Non-rigid · Graph Matching · Quadratic Assignment · Line feature

1 Introduction

Finding correspondences between two sets of spatial features is one of the fundamental problems in com-

Hang Dou · Tao Ju
Washington University in St. Louis

Matthew L Baker
Baylor College of Medicine

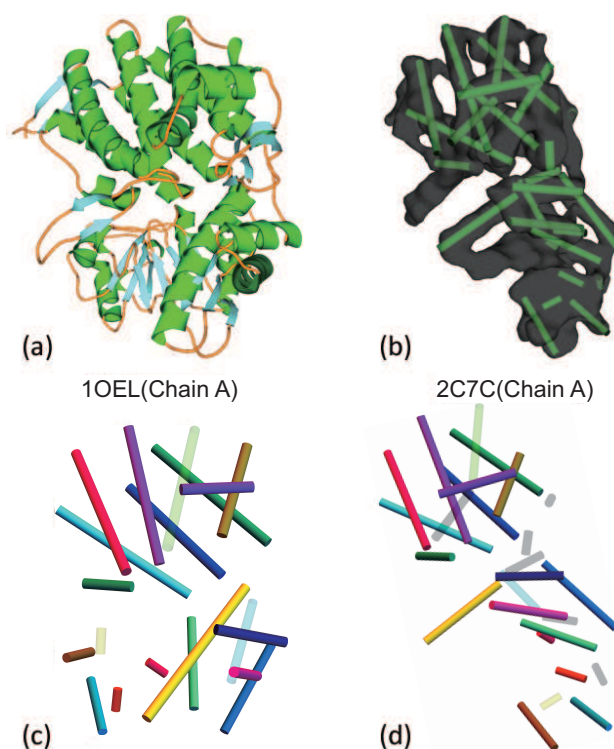


Fig. 1 Our work is motivated by the problem of deforming a protein structure (a) into a density volume (b) guided by the correspondences between the helices (green spirals in (a) and green rods in (b)). The desired matching is shown at the bottom (helices in two proteins that do not have well-defined correspondences are shown in transparency).

puter graphics and pattern recognition. The problem is particularly challenging if the features are related by a non-rigid deformation. In this paper, we consider a line feature matching problem that is motivated by a biological application.

One of the challenges in understanding the 3D structure of a protein, a unique shape made up of a chain

of amino acids, is that the protein may exhibit a different shape in different states. While the structure of a protein is easier to determine in certain state (e.g., when the protein is isolated and crystallized), it is much harder in others (e.g., when the protein is embedded in a macromolecular complex). Oftentimes only low-resolution intensity volumes are available to capture the protein in a non-isolated state [7]. A common strategy to recover protein structures is therefore to *deform* the structure known at one state (e.g., Figure 1 (a)) to *fit* the intensity volume capturing the protein at another state (e.g., Figure 1 (b)) [22,23].

As part of a larger effort, we are investigating more accurate and efficient means for protein fitting. Our effort is motivated by the fact that α -helices (green spirals in Figure 1 (a)) are usually stable under state changes, and that potential locations of helices in the intensity volume (green rods in Figure 1 (b)) can be robustly identified using a variety of methods [27,3]. Hence the idea is to guide protein fitting by first computing the correspondences between the two sets of helices. The correct correspondence (as shown in Figure 1 (c,d)) should match up helices with similar lengths, and the deformation that aligns the corresponding helices should “bend” the protein as little as possible.

While point features have been extensively studied in the past for registration [14,20], non-affine matching of line segments has received little attention so far (with the exception of [1]). The naive extension of point-based methods to register line segments, by treating each segment as a point associated with properties like length and orientation, will not work since the orientation of a segment only makes sense when it is considered in the context of other segments. Determining the orientation of the segments is further complicated by the fact that our segments are *undirected*.

Contributions We present a novel graph-based method for matching two sets of undirected 3D line segments which under go non-rigid deformation. Such deformation may alter the position, length and direction of each line segment. We make two technical contributions:

1. We formulate line segment matching as an affinity-based graph-matching problem. By splitting an undirected segment into two directed segments and appropriately setting up the affinity functions, our graph construction captures two goals: matching segments with similar lengths and undergoing an as-rigid-as-possible deformation.
2. We solve the graph matching problem using continuous relaxation. In particular, we propose a novel technique in the final step where a continuous score

vector (which describes the confidence of every pair-wise assignment) is discretized into a matching. Compared with existing discretization techniques [12,15,16], our method produces more confident assignments and is applicable to problems with more general mapping constraints (such as ours).

When tested on both synthetic inputs and authentic protein data, our method outperforms existing methods both for discretization in graph matching and for registering line segments.

2 Related work

While there is a sizable literature on feature registration (see surveys [14,20]), we briefly review works on non-rigid registration and particularly those that are relevant to our graph-base approach.

The vast majority of registration methods are designed for points associated with attributes (e.g., SIFT in image analysis, surface normal and local frames on surfaces, etc.). The objective is to maximize the attribute similarity between the matching features while minimizing the distortion in the deformation that aligns the matching features. Due to the expense in computing the deformation, the distortion term is usually expressed between two pairs of features, such as the difference between the pair-wise Euclidean [15,5] distances, geodesic [2] distances or feature strength [11]. Hence the problem of finding correspondences can be formulated as a graph matching problem that maximizes the sum of node affinity (capturing attribute similarity) and edge affinity (capturing distortion) over all matched node pairs and edge pairs. Note that more complex forms of distortion can be formulated between triples or n-tuples of features, but optimization is significantly more complex [6,10].

Graph matching, which can be casted as a quadratic assignment (QA) problem, is known to be NP-hard [17]. Techniques for solving graph matching generally fall into two categories. The first group of methods apply combinatorial optimization, such as branch-and-bound [24], probabilistic methods [2], importance sampling [21], and ant-colony optimization [25]. However, these methods usually have high computational cost. The second group (to which our method belongs) approximates the matching by a continuous solution that computes, for each possible assignment between a pair of features, a confidence score. This can be done efficiently using a variety of methods, such as semi-definite programming [18], graduated assignment [12], spectral matching [15,9], and relaxation labelling [30]. The continuous scores are then discretized to form the matching, most commonly by a simple filtering scheme based

on the scores [15]. This initial matching can be further improved by following directions of increasing affinity [16, 12]. These techniques iteratively seek better matchings by solving a linear assignment problem at each step, which can be done efficiently for certain types of matchings, such as one-to-one or many-to-one.

There are much fewer studies on the matching of line segments. Most of the methods are developed for stereo matching [13, 8, 26], where the deformation between the two sets of segments is restricted to an affine transformation. Also, image features are usually available as additional features to aid the matching.

To our best knowledge, the only work that performs non-affine matching between 3D line segments is done by Abeyasinghe et al. [1]. Unlike our method, Abeyasinghe’s algorithm is designed for deformations that can be represented as piece-wise near-rigid transformations. The algorithm represents isometry between clusters of line segments as cliques in an associate graph whose nodes are pair-wise assignments, and searches for the largest and most coherent cliques using branch-and-bound. While producing promising results, Abeyasinghe’s algorithm often fails when the isometric clusters become too small. This happens when the desired deformation is no longer piece-wise rigid, but fully non-rigid, a phenomenon that is not uncommon for proteins undergoing large shape changes.

3 Graph-based line matching

The desirable correspondences between two sets of line segments should match segments with similar lengths, and the deformation implied by the matching should have a low non-rigidity, or as rigid as possible. As the first step, we cast the problem as that of graph matching. With appropriate construction of the graph structure and definition of affinities, we can capture both our matching goals, namely similarity in segment lengths and rigidity in deformation, as the optimal matching between two graphs.

3.1 Affinity-based graph matching

We first review the general formulation of a graph matching problem. Consider two complete graphs S, T with node sets V_S, V_T (assuming that $\|V_S\| \leq \|V_T\|$). We are also given a *node affinity* function $f_n(\{s, t\})$ that measures the similarity between node $s \in V_S$ and node $t \in V_T$, as well as an *edge affinity* function $f_e(\{s, t\}, \{s', t'\})$ that measures the similarity between the edge connecting $s, s' \in V_S$ and the edge connecting $t, t' \in V_T$.

Graph matching seeks a mapping $X : V_S \rightarrow V_T$ that is injective (meaning $X(s) \neq X(s')$ if $s \neq s'$) and

maximizes the total affinity:

$$\sum_{s \in V_S} f_n(\{s, X(s)\}) + \sum_{s \in V_S} \sum_{s' \in V_S} f_e(\{s, X(s)\}, \{s', X(s')\}) \quad (1)$$

3.2 Setting up the graph

A natural choice of S and T for our problem would have one node for each line segment in the source or target set. The node affinity $f_n(\{s, t\})$ would measure the similarity in length between the two segments s and t . The edge affinity $f_e(\{s, t\}, \{s', t'\})$ would measure the isometry of the most rigid deformation that brings segments pair s, s' into alignment with pair t, t' .

The problem with this choice is that a matching with a high total affinity may not represent a matching with high rigidity. This is because pair-wise isometry between pairs of *undirected* segments may not imply isometry between larger groups of segments. This problem is illustrated by the simple example in Figure 2 (top), where the source and target sets have respectively 3 and 4 segments. While each segment pair s_i, s_j in the source for $i, j = 1, 2, 3$ can be rigidly aligned with the pair t_i, t_j in the target, there is no isometry between the two groups of segments s_1, s_2, s_3 and t_1, t_2, t_3 . As a result, the matching $X(s_i) = t_i$ for $i = 1, 2, 3$ would have the same total affinity as the alternative matching where $X(s_3) = t_4$, even though the latter represents a truly rigid (and hence desirable) matching.

On the other hand, it was shown that pair-wise isometry between pairs of *directed* segments imply isometry of whole groups of segments [1]. Hence we impose directions on the input segments to create our graph. Due to symmetry, we arbitrarily assign a direction to each segment in the source set, and split each segment in the target set into two with opposite directions. Each directed segment is then represented by a node in the corresponding graph. Hence the graph S will have the same number of nodes as the number of source line segments, whereas the graph T will have twice the number of nodes as the number of target line segments. We call two nodes representing the same undirected segment *antinodes* of each other and denote them by t, \bar{t} (so that $\bar{\bar{t}} = t$). The graph nodes for the input in Figure 2 (top) are shown in the bottom. Observe that the only matching that involves perfect pair-wise isometry is the desirable mapping $X(s_1) = t_1, X(s_2) = t_2, X(s_3) = t_4$.

With the new graph construction, we need to tighten the injectivity condition of the mapping X . In particular, among the two nodes in T that represent the same undirected line segment, at most one could be the image of X . To summarize,

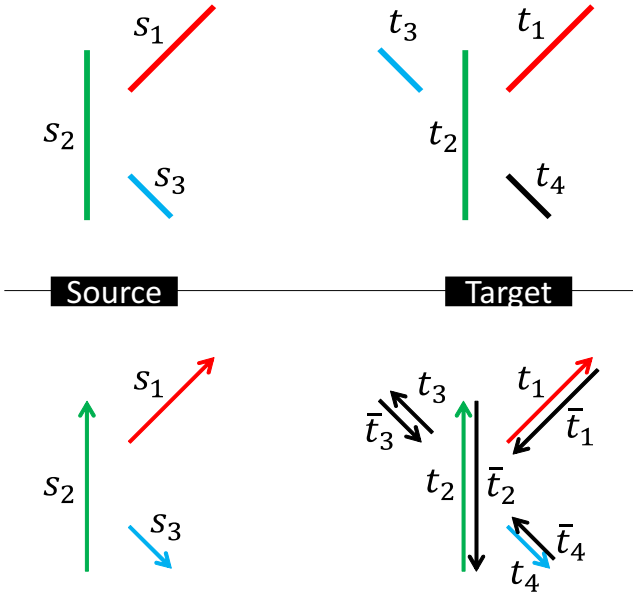


Fig. 2 Comparing two graph representations: one node for each undirected line segment (top), or one node for each directed line segment after imposing directions on the input (bottom). The matching with the highest total affinity is indicated by the colors.

Properties 1. *Mapping X needs to satisfy:*

1. $\forall s \in S, X(s)$ exists and is unique.
2. $\forall s, s' \in S$ such that $s \neq s', X(s) \neq X(s')$ and $X(s) \neq \overline{X(s')}$.

3.3 Defining the affinities

With the graph constructed, we next detail the definition of the affinity functions. Naturally, node affinity measures the similarity between two line segments. Let \mathbf{s} be the directed line segment represented by node s , we define

$$f_n(\{s, t\}) = w(\|\mathbf{s}\| - \|\mathbf{t}\|, \sigma_l),$$

where σ_l is a user-controlled parameter and w is the Gaussian function

$$w(r, \sigma) = \exp\left(-\frac{r^2}{2\sigma^2}\right).$$

To capture the isometry between two pairs of directed line segments, we follow a similar approach to that in [1], which allows the user to fine-tune the contribution of different types of non-isometry. Specifically, we first capture the relative geometric relation between two source (or target) segments, \mathbf{s}, \mathbf{s}' , by six measures: 1) the lengths of the segments ($\|\mathbf{s}\|, \|\mathbf{s}'\|$), 2) the distance between their mid-points, denoted as $\text{dis}(\mathbf{s}, \mathbf{s}')$, and 3) a set of angles denoted as $\text{ang}(\mathbf{s}, \mathbf{s}') = \{\alpha, \beta, \theta\}$

where θ is the angle from \mathbf{s} to \mathbf{s}' , α (resp. β) is the angle from \mathbf{s} (resp. \mathbf{s}') to the vector connecting the midpoint of \mathbf{s} (resp. \mathbf{s}') to the midpoint of the other segment. These measures are illustrated in Figure 3. These measures determine a unique pair of directed segments, and they are invariant to translation and rotation.

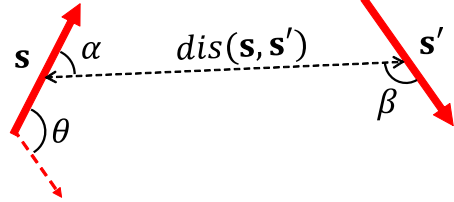


Fig. 3 Six measures that capture the geometric relationship between two directed line segments \mathbf{s}, \mathbf{t} .

The edge affinity $f_e(\{s, t\}, \{s', t'\})$ measures the similarity between the geometric measures of the pair \mathbf{s}, \mathbf{s}' and those of the pair \mathbf{t}, \mathbf{t}' . Note that the two sets of measures are identical if and only if the two pairs of segments are related by an isometry. While one could measure the similarity between all segment pairs with equal importance, such edge affinity could over-penalize deformations that are near-rigid locally but highly non-rigid as a whole (e.g., bending a co-linear sequence of line segments into a curve). For applications that allow such deformations (e.g., protein fitting), it would be ideal to downplay the isometry between segment pairs that are far away. As a result, we define

$$f_e(\{s, t\}, \{s', t'\}) = w(\|\mathbf{s}\| - \|\mathbf{t}\|, \sigma_l) \times w(\|\mathbf{s}'\| - \|\mathbf{t}'\|, \sigma_l) \\ \times w(\text{dis}(\mathbf{s}, \mathbf{s}') - \text{dis}(\mathbf{t}, \mathbf{t}'), \sigma_d) \\ \times w(\text{ang}(\mathbf{s}, \mathbf{s}') - \text{ang}(\mathbf{t}, \mathbf{t}'), \sigma_a) \\ \times w(\text{dis}(\mathbf{s}, \mathbf{s}'), \sigma_w) \times w(\text{dis}(\mathbf{t}, \mathbf{t}'), \sigma_w).$$

Here, $\sigma_l, \sigma_d, \sigma_a, \sigma_w$ are user-provided parameters. The last two terms on the right-hand side act as a distance-based attenuation to downplay the isometry between far-away pairs.

Our approach to measure the isometry between segment pairs is similar to that in [1], and particularly, the same set of six measures are used in both works to capture the geometric relation between a pair of directed segments \mathbf{s}, \mathbf{s}' . The method of [1] then applies thresholding for each of the six measures to make a *binary* decision of whether this pair is isometric with another pair \mathbf{t}, \mathbf{t}' . Such thresholding not only makes the method highly sensitive to the choice of thresholds but also limits the method to inputs that are made up of clusters of segments undergoing near-rigid transformations (so called *semi-rigid* deformations). In contrast, by computing a continuously ranged score (f_e), our method is

less sensitive to parameters and can handle more general, fully non-rigid deformations (see our results).

4 Solving graph matching

Algorithms for solving affinity-based graph matching typically proceeds in two steps. First, a continuous form of the mapping X is sought that minimizes the total affinity. Second, the continuous mapping is discretized while observing the mapping constraint (e.g., injectivity). We follow the same paradigm, but proposes a novel approach for the second step. Our discretization algorithm improves upon a simple greedy algorithm that has been commonly employed in the past. Compared with other discretization methods [12, 16], our algorithm can be applied to more general mapping constraints (e.g., Properties 1).

4.1 Quadratic assignment formulation

We start by formulating graph matching as a quadratic assignment (QA) problem, so that it is amendable to a continuous solution. The QA formulation is common in previous graph matching methods, but we include it here for completeness.

We first encode the affinity functions in a single affinity matrix M with dimension $n_S n_T \times n_S n_T$, where n_S, n_T are respectively the number of nodes in the graph S, T . Each column (or row) of M represents an *assignment* $a = \{s, t\}$ between a node $s \in V_S$ and node $t \in V_T$. The diagonal and off-diagonal elements of M are respectively the node and edge affinity, that is,

$$M(a, b) = \begin{cases} f_n(a) & \text{if } a = b \\ f_e(a, b) & \text{otherwise} \end{cases}$$

where a, b are the assignments represented by the corresponding column and row. Note that, due to our definition of f_e , M is symmetric.

The mapping X can be represented by an *assignment vector* \mathbf{x} whose length is $n_S n_T$ and whose elements are either 0 or 1. Properties 1 can be encoded by the following equality and inequalities for appropriate choices of constant matrices C, D :

$$C\mathbf{x} = \mathbf{1}, D\mathbf{x} \leq \mathbf{1}. \quad (2)$$

The problem of finding an optimal mapping between the two graphs becomes seeking an assignment vector \mathbf{x} that maximizes the following form of total affinity (equivalent with Equation 1) subject to Equation 2:

$$\mathbf{x}^T M \mathbf{x}. \quad (3)$$

4.2 Continuous-discrete matching

We briefly review the continuous-discrete paradigm to solve the QA problem formulated above. In the first step, the maximization of Equation 3 is solved by dropping the integer requirement of the assignment vector and relaxing the mapping constraints. While several methods exist for this step (as reviewed in Section 2), we adopt the spectral method introduced by Leordeanu and Hebert [15]. By completely dropping the mapping constraint, they seek a continuous vector $\tilde{\mathbf{x}}$ that maximizes $\frac{\tilde{\mathbf{x}}^T M \tilde{\mathbf{x}}}{\tilde{\mathbf{x}}^T \tilde{\mathbf{x}}}$. By Raleigh's ratio theorem, the maximizing vector $\tilde{\mathbf{x}}$ is the principal eigenvector of M . Each value $\tilde{\mathbf{x}}(a)$ for an assignment a can be regarded as an indication of how likely a is included in the matching. In the second step, the continuous vector $\tilde{\mathbf{x}}$ is discretized to an assignment vector that satisfies the mapping constraints. The common way to do this, which was proposed also in [15], is iteratively assigning 1s to assignments whose corresponding values are high in $\tilde{\mathbf{x}}$ while avoiding violation of the mapping constraints. Specifically, starting with the zero vector $\mathbf{x} = \mathbf{0}$, we visit each assignment a by decreasing value of $\tilde{\mathbf{x}}(a)$. We set $\mathbf{x}(a) = 1$ if $C\mathbf{x} \leq \mathbf{1}$ and $D\mathbf{x} \leq \mathbf{1}$, otherwise skip to the next assignment. We stop when all sources nodes have found their matches, or when $C\mathbf{x} = \mathbf{1}$.

The two-step algorithm is presented in the pseudo-code of Algorithm 1. Here, **ContinuousMatch**(M, C, D) can be any method that computes a continuous vector, and **Binarize**($\tilde{\mathbf{x}}, C, D$) refers to the simple discretization strategy above.

Algorithm 1 Match(M, C, D)

```

 $\tilde{\mathbf{x}} := \text{ContinuousMatch}(M, C, D)$ 
 $\mathbf{x} := \text{Binarize}(\tilde{\mathbf{x}}, C, D)$ 
return  $\mathbf{x}$ 

```

4.3 Better discretization

As observed in earlier works [16], the simple discretization method often creates suboptimal assignment vectors. The remedies proposed so far [12, 16] adopt a gradient-descent strategy. Starting from an initial assignment vector, a better vector is chosen by walking in the gradient direction of the total affinity (Equation 3). Such strategy is quite efficient, as it requires solving only a linear assignment problem at each step, which can be done quickly and optimally for certain class of mappings (e.g., injective mappings). However, we do not know of any efficient and optimal solutions for linear assignment with our novel mapping constraints (Properties 1). Moreover, as we shall see in the results, gradient-

descent can be trapped in local minima and hence sensitive to the quality of the initial assignment vector.

We present here a novel discretization method that outperforms the simple method above and is applicable to our mapping constraints. Even for problems that require only injective mappings (and hence gradient-descent approaches can be used [12,16]), our method can offer a better starting point for gradient-descent (see our results).

We observed that the continuous solution $\tilde{\mathbf{x}}$ often only contains a few large values, which typically capture those obviously good assignments, and many low and similar values. Hence it is difficult and error-prone to pick out the less-obvious assignments just by values of $\tilde{\mathbf{x}}$. On the other hand, the less-obvious assignments can become more obvious if we fix a few assignments that we are already confident about, since the problem is now constrained by the deformation determined from the fixed assignments. Hence our method proceeds by iteratively formulating smaller QA problems, each constrained by the confident assignments found in the previous, larger problem, until all source nodes have found their matches.

We first slightly modify the simple discretization method (**Binarize**($\tilde{\mathbf{x}}, C, D$)) to pick up only assignments with high values in $\tilde{\mathbf{x}}$. Instead of terminating the process only when we fulfill the mapping constraint or $C\mathbf{x} = \mathbf{1}$, we also stop if the current assignment a being considered has a low confidence, or $\tilde{\mathbf{x}}(a) < \max(\tilde{\mathbf{x}}) * r$ where $r \in [0, 1]$ is user-given parameter. We denote this process as **SelectiveBinarize**($\tilde{\mathbf{x}}, C, D, r$). Note that the new procedure reduces to the original simple method when $r = 0$.

Given an assignment vector \mathbf{x} that partially fulfills the mapping constraints (i.e., $C\mathbf{x} \leq \mathbf{1}$), we formulate a reduced QA problem for those nodes in the two graphs such that neither the node nor its antinode appears in the assignments in \mathbf{x} . Specifically, let us denote by A the set of assignments that either appear in \mathbf{x} or whose addition to \mathbf{x} would violate the mapping constraints of Equation 2. We construct smaller affinity matrix, $M_{\mathbf{x}}$, and constraint matrices $C_{\mathbf{x}}, D_{\mathbf{x}}$, for the reduced problem as follows:

- $C_{\mathbf{x}}$ (resp. $D_{\mathbf{x}}$): remove any row of C (resp. D) whose product with \mathbf{x} equals 1 and any column of C (resp. D) representing assignments in A .
- $M_{\mathbf{x}}$: remove any row or column of M representing assignments in A , and set in the remaining submatrix

$$M_{\mathbf{x}}(a, b) = \begin{cases} M(a, a) + 2M(a, \cdot)\mathbf{x} & \text{if } a = b \\ M(a, b) & \text{otherwise} \end{cases}$$

The addition to the values along the diagonal of M , as shown above, acts as the influence of the already-chosen assignments in \mathbf{x} on the solution of the reduced problem. We denote the reduction process as $\{M_{\mathbf{x}}, C_{\mathbf{x}}, D_{\mathbf{x}}\} = \mathbf{Reduce}(M, C, D, \mathbf{x})$.

We can show that our reduction of the problem maintains the optimality of the solution, if the assignments in \mathbf{x} are part of the optimal matching. Our statement holds for any symmetric affinity matrix M and constraint matrices C, D . Given two assignment vectors \mathbf{y}, \mathbf{z} for reduced problems of different sizes, we denote by $\mathbf{y} \oplus \mathbf{z}$ the assignment vector that includes the union of assignments from both vectors.

Proposition 1 *Let $\{M_{\mathbf{x}}, C_{\mathbf{x}}, D_{\mathbf{x}}\} = \mathbf{Reduce}(M, C, D, \mathbf{x})$ for some assignment vector \mathbf{x} such that $C\mathbf{x} \leq \mathbf{1}, D\mathbf{x} \leq \mathbf{1}$. If assignment vector \mathbf{z} is optimal for the reduced problem (i.e., maximizing $\mathbf{z}^T M_{\mathbf{x}} \mathbf{z}$ subject to $C_{\mathbf{x}} \mathbf{z} = \mathbf{1}$ and $D_{\mathbf{x}} \mathbf{z} \leq \mathbf{1}$), then the assignment vector $\mathbf{y} = \mathbf{z} \oplus \mathbf{x}$ is optimal for the original problem (i.e., maximizing $\mathbf{y}^T M \mathbf{y}$ subject to $C\mathbf{y} = \mathbf{1}$ and $D\mathbf{y} \leq \mathbf{1}$) among all \mathbf{y} that includes the assignments in \mathbf{x} .*

Proof. It is easy to show that if \mathbf{z} meets the mapping constraints ($C_{\mathbf{x}} \mathbf{z} = \mathbf{1}$ and $D_{\mathbf{x}} \mathbf{z} \leq \mathbf{1}$) then so does \mathbf{y} ($C\mathbf{y} = \mathbf{1}$ and $D\mathbf{y} \leq \mathbf{1}$), and vice versa. We will only show here that maximizing $\mathbf{z}^T M_{\mathbf{x}} \mathbf{z}$ is equivalent to maximizing $\mathbf{y}^T M \mathbf{y}$. In fact,

$$\begin{aligned} \mathbf{y}^T M \mathbf{y} &= \mathbf{x}^T M \mathbf{x} + \mathbf{z}^T M \mathbf{z} + \mathbf{z}^T M \mathbf{x} + \mathbf{x}^T M \mathbf{z} \\ &= \mathbf{x}^T M \mathbf{x} + \mathbf{z}^T M \mathbf{z} + 2\mathbf{z}^T M \mathbf{x} \\ &= \mathbf{x}^T M \mathbf{x} + \mathbf{z}^T M_{\mathbf{x}} \mathbf{z} \end{aligned}$$

The second equality is due to symmetry of M , and the last equality comes from our construction of $M_{\mathbf{x}}$. Since \mathbf{x} is given, this shows that the same \mathbf{z} that maximizes the affinity in the reduced problem also maximizes the affinity in the original problem, among all solutions that contain the assignments in \mathbf{x} . \square

Our algorithm then involves iterative reduction of the problem, each time obtaining assignments of high values in the continuous solution and using them to constrain the smaller problems. The pseudo-code is shown in Algorithm 2.

Our algorithm has a single parameter r , the cut-off value in the continuous vector $\tilde{\mathbf{x}}$ above which the discrete assignments are obtained. This parameter controls the trade-off between the confidence of the solution and computational efficiency. A larger value of r will result in fewer, more confident assignments being selected at each iteration, but there will be more iterations. On the other hand, a smaller r will lead to faster reduction of the problem sizes, albeit at the cost of producing

less confident assignments. When $r = 0$, our **IterativeMatch** algorithm reduces to the classical **Match** algorithm. Note that our algorithm always terminates, since the number of iterations cannot exceed n_S . However, like **Match**, our **IterativeMatch** is not guaranteed to return an optimal solution.

Algorithm 2 **IterativeMatch**(M, C, D, r)

```

 $\tilde{\mathbf{x}} := \text{ContinuousMatch}(M, C, D)$ 
 $\mathbf{x} := \text{SelectiveBinarize}(\tilde{\mathbf{x}}, C, D, r)$ 
if  $C_{\mathbf{x}} \neq \mathbf{1}$  then
   $\{M_{\mathbf{x}}, C_{\mathbf{x}}, D_{\mathbf{x}}\} = \text{Reduce}(M, C, D, \mathbf{x})$ 
   $\mathbf{x} = \mathbf{x} \oplus \text{IterativeMatch}(M_{\mathbf{x}}, C_{\mathbf{x}}, D_{\mathbf{x}}, r)$ 
end if
return  $\mathbf{x}$ 

```

5 Results

We tested our algorithm on line segments representing protein helices and compared with previous methods for graph matching and line segment matching. We used two groups of test data, one deforming a single protein synthetically to create targets with varying level of similarity with the source, and the other featuring pairs of authentic proteins. In all experiments, we used a fixed setting $\{\sigma_a, \sigma_d, \sigma_l, \sigma_w\} = \{90^\circ, 5\text{\AA}, 40\text{\AA}, 120\text{\AA}\}$ for calculating the affinities.

5.1 Synthetic targets

We use the set of helices from one protein (3tgl, chain A) as our source and follow the procedure in [15] to generate target sets using thin-plate spline (TPS) deformation at varying level of bending energy. Control points of TPS are placed on a 3D lattice and are displaced to induce a deformation on the end points of the line segments. We randomly generate 50 trials for each level of bending energy and report the average. Examples of the source and TPS targets at different bending energy are shown in Figure 4 (a).

Our iterative discretization method (**IterativeMatch**) consistently performs better than the simple discretization approach of [15] (**Match**), both in terms of total affinity of the assignment vector (Figure 4 (b)) and the matching accuracy (Figure 4 (c)). In Figure 4 (b), the affinities for **IterativeMatch** are plotted as ratios over the affinity for **Match**. In Figure 4 (c), the accuracy is measured as the proportion of source segments that are mapped to the correct target segments. Also, observe that larger values of r generally result in higher affinity and matching accuracy.

On the other hand, the matching accuracy of the semi-rigid matching algorithm [1] is significantly lower than graph matching as the bending energy increases

(Figure 4 (c)). This demonstrates that the algorithm, which was designed for deformations made up of piecewise near-rigid transformations, cannot handle well a fully non-rigid deformation.

We further compare our iterative method for discretization with the gradient-descent approach for improving an initial assignment vector [16]. Since the latter only applies to injective mappings, we modified our problem to match two sets of *directed* line segments. The graphs are constructed in the same way as in the undirected case, except that there are no anti-nodes in the target graph T . We used the assignment vector computed by either the simple discretization method or our iterative algorithm as the initial vector and applied gradient descent.

The results (Figure 4 (d,e)) indicated that the gradient descent method is highly sensitive to the quality of the initial assignment. While it was able to slightly improve upon the result of the simple discretization method, the improvement was rather small compared to the result of our iterative discretization algorithm. On the other hand, we did not observe any further improvement to the result of our algorithm using gradient descent.

5.2 Authentic targets

We next validated our method in matching helices from two authentic proteins. We selected 16 pairs of proteins from the Protein Data Bank [4], which have been used extensively in the computational biology community to evaluate protein fitting methods [28, 29]. Each pair consists of either the same proteins at two different states or two proteins with highly similar primary sequences, so that the ground truth matching between their helices can be obtained from their primary sequences. It is possible for some helices in one protein to have no exact match in the other protein. We consider such helices as “noise” in the input. While we compute matching for all source helices, including noise, we exclude noise helices when measuring matching accuracy.

Figure 5 compares our method with **Match** and semi-rigid matching [15] on the 16 protein pairs. Note even for values of r as low as 0.3, our method outperforms both simple discretization method of [15] and the semi-rigid matching algorithm of [1], achieving 100% matching accuracy in all 16 pairs (Figure 5 (b)). A few example protein pairs are shown in Figure 6, where we mark the incorrectly matched helices of the other two methods. Note that the deformation of the source protein is often complex and far from being piecewise rigid. Also, the semi-rigid algorithm often leaves several target helices unmatched due to its use of thresholding.

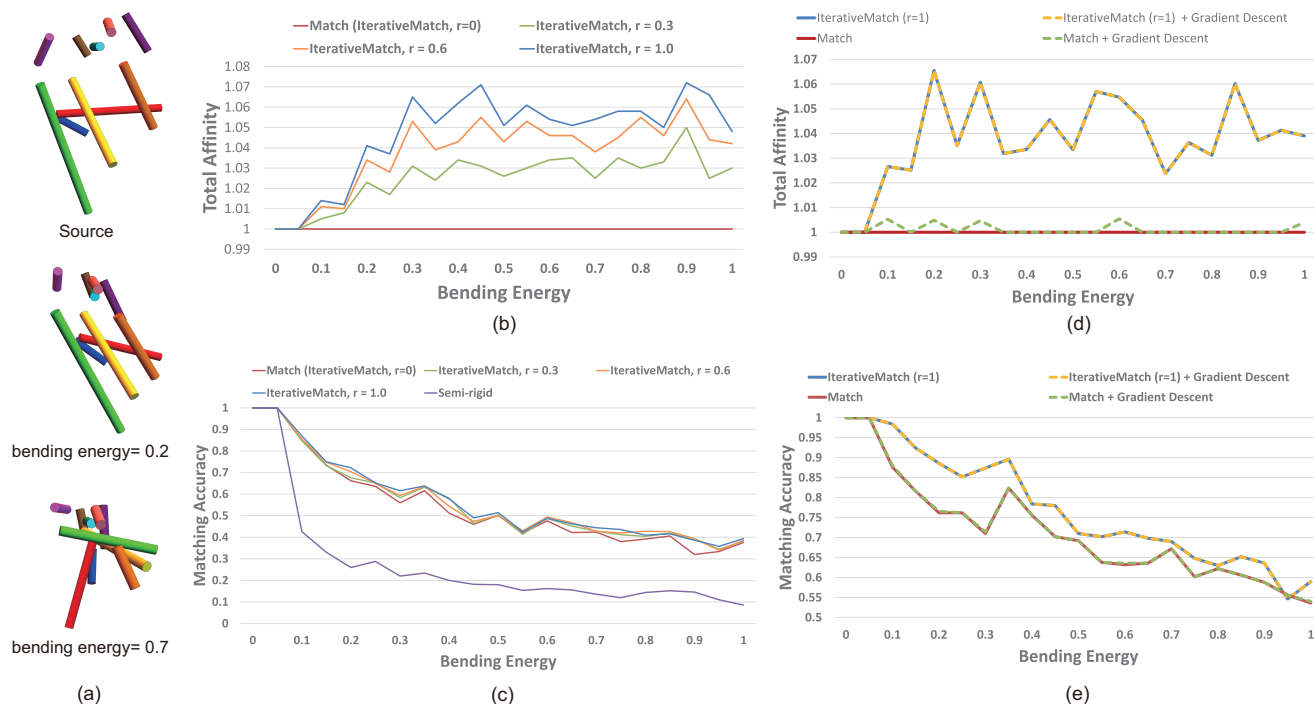


Fig. 4 Matching synthetically created targets: (a) the source segments (top) and two examples of targets deformed from the source using TPS with small and large levels of bending energy, (b) ratios of the total affinity of matchings found by our iterative discretization method (**IterativeMatch**) with different r values over the total affinity of the simple discretization method (**Match**), (c) accuracy of matchings found by these discretization methods, (d,e) affinity and accuracy of matchings found by **IterativeMatch** ($r = 1$) and **Match** with and without gradient descent improvement in a modified input where the segments are oriented.

Protein Pairs	Total Helices	Building M	IterativeMatch		
			$r = 0.6$	$r = 0.3$	$r = 0$
1j4p-A : 1k3q-A	6	0.	0.183	0.201	0.184
1xve-F : 1mty-G	16	0.002	0.186	0.189	0.165
4ake-A : 1ake-A	18	0.003	0.186	0.211	0.149
3e8k-A : 2gp1-A	19	0.002	0.188	0.176	0.164
3e8k-A : 3ddx-A	19	0.002	0.182	0.190	0.149
3tgl-A : 4tgl-A	20	0.006	0.176	0.195	0.152
2lao-A : 1lst-A	20	0.004	0.171	0.160	0.147
1vfv-A : 1i5s-A	24	0.008	0.195	0.186	0.148
1omp-A : 1anf-A	33	0.036	0.191	0.174	0.186
1aon-A : 1aon-H	40	0.052	0.237	0.209	0.217
1oel-A : 1ss8-A	41	0.077	0.238	0.220	0.212
1ss8-A : 1sx4-A	42	0.059	0.251	0.217	0.210
1oel-A : 2c7c-A	46	0.097	0.334	0.269	0.242
1lfg-A : 1lfh-A	50	0.150	0.314	0.281	0.293
1n0u-A : 1n0v-C	71	0.475	1.114	1.077	0.997
1oao-C : 1oao-D	76	0.697	1.524	1.407	1.141

Table 1 Timing of our method (in seconds).

Performance Our algorithm is implemented in C++ and tested on a machine with a 3.60GHz CPU (Intel(R) Core(TM) i7-4960X) and 16GM memory. We used Matlab for finding the principle eigenvector of the affinity matrix (needed for **ContinuousMatch**). Our implementation found the matching for each test protein pair within seconds. A detailed break-down of run time between graph construction (computing the affin-

ity matrix M) and graph matching (**IterativeMatch**) for different r values is shown in Table 1. Note that the addition in computational time with larger r is not very significant, which indicates that the problem size (i.e., dimension of M) is reduced quickly during our algorithm.

6 Conclusion

We present an algorithm for finding correspondences between two sets of 3D line segments that are related by some non-rigid deformation. Our novelty lies in the formulation of a graph matching problem that captures the desired correspondences as a matching with maximal affinity, and in solving the matching problem using an iterative continuous-discrete paradigm that produces better solutions than previous discretization techniques used in continuous relaxation. Our algorithm is validated on protein data sets and showed improved accuracy over the state-of-art method for the same problem.

A potential issue when applying our iterative discretization method to larger inputs is its computational cost when the value of r is high, since it needs to solve many QA problems. A possible strategy would be to use a variable r that starts small, allowing the size of

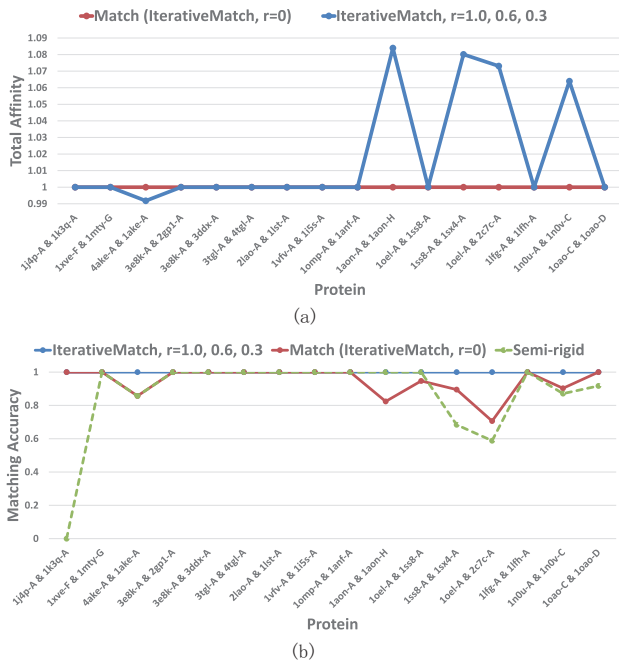


Fig. 5 Matching authentic protein pairs: (a) ratios of the total affinity of matchings found by **IterativeMatch** with different r over the total affinity of matchings found by **Match** for all 16 protein pairs, (b) accuracy of matchings of various methods.

the QA problem to quickly reduce, and increases with more iterations.

Our algorithm seeks to maximize the total affinity between two feature sets and thus cannot handle partial matching as it is. However, it is possible to achieve partial matching by pruning the correspondences in a post-process, as done similarly in [19].

7 Acknowledgements

The work is supported in part by NSF grants (DBI-1356388, DBI-1356306) and NIH grants (5P41GM103832, 2R01GM079429, R21GM100229).

References

1. S. S. Abeyasinghe, M. L. Baker, W. Chiu, and T. Ju. Semi-isometric registration of line features for flexible fitting of protein structures. *Comput. Graph. Forum*, 29(7):2243–2252, 2010.
2. D. Anguelov, D. Koller, P. Srinivasan, S. Thrun, H.-C. Pang, and J. Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Advances in Neural Information Processing Systems (NIPS 2004)*, Vancouver, Canada, 2004.
3. M. L. Baker, T. Ju, and W. Chiu. Identification of secondary structure elements in intermediate-resolution density maps. *Structure*, 15(1):7 – 19, 2007.
4. H. Berman, K. Henrick, and H. Nakamura. Announcing the worldwide protein data bank. *Nature Structural & Molecular Biology*, 10(12):980–980, 2003.

5. W. Chang and M. Zwicker. Range scan registration using reduced deformable models. *Comput. Graph. Forum*, 28(2):447–456, 2009.
6. M. Chertok and Y. Keller. Efficient high order matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(12):2205–2215, 2010.
7. W. Chiu, M. L. Baker, and S. C. Almo. Structural biology of cellular machines. *Trends Cell Biol.*, 16:144–150, Mar 2006.
8. S.-L. Chou and W.-H. Tsai. Line segment matching for 3d computer vision using a new iteration scheme. *Mach. Vis. Appl.*, 6(4):191–205, 1993.
9. T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press, 2007.
10. O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2383–2395, 2011.
11. W. Feng, J. Huang, T. Ju, and H. Bao. Feature correspondences using morse smale complex. *The Visual Computer*, 29(1):53–67, 2013.
12. S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.
13. R. P. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1168–1180, Nov. 1989.
14. O. V. Kaick, H. Zhang, G. Hamarneh, and D. Cohen-or. A survey on shape correspondence, 2011.
15. M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1482–1489. IEEE, 2005.
16. M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Advances in Neural Information Processing Systems*, pages 1114–1122, 2009.
17. E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657 – 690, 2007.
18. C. Schellewald and C. Schnorr. Probabilistic subgraph matching based on convex relaxation. In *EMMCVPR*, volume 3757 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 2005.
19. G. K. Tam, R. R. Martin, P. L. Rosin, and Y.-K. Lai. Diffusion pruning for rapidly and robustly selecting global correspondences using local isometry. *ACM Trans. Graph.*, 33(1):4, 2014.
20. G. K. L. Tam, Z. quan Cheng, Y. kun Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. fang Sun, and P. L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to non-rigid.
21. A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H.-P. Seidel. Isometric registration of ambiguous and partial data. In *CVPR*, pages 1185–1192. IEEE, 2009.
22. M. Topf, K. Lasker, B. Webb, H. Wolfson, W. Chiu, and A. Sali. Protein structure fitting and refinement guided by cryo-em density. *Structure*, 16(2):295 – 307, 2008.
23. L. G. Trabuco, E. Villa, K. Mitra, J. Frank, and K. Schulten. Flexible fitting of atomic structures into electron microscopy maps using molecular dynamics. *Structure*, 16(5):673 – 683, 2008.

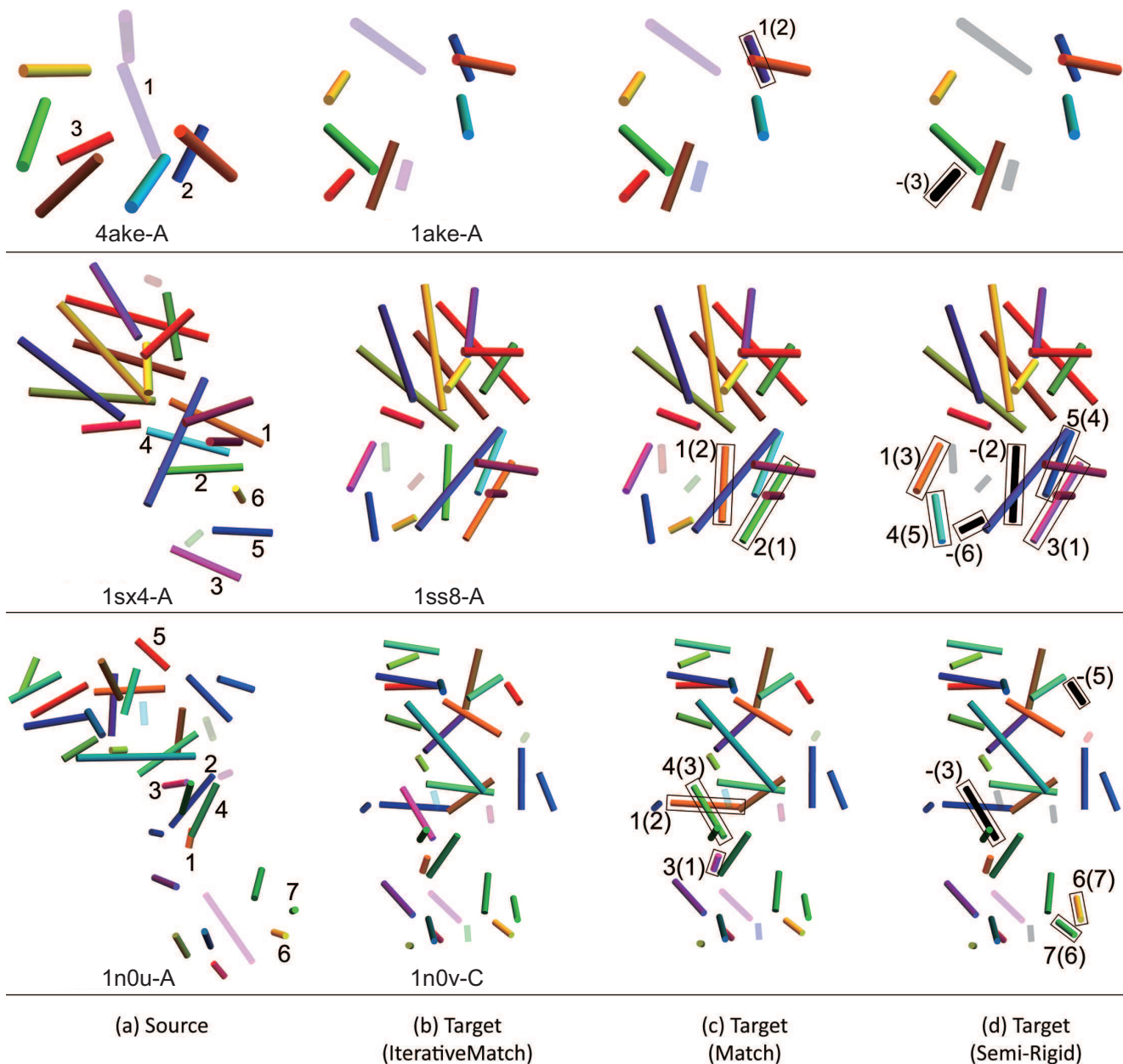


Fig. 6 Examples of matching protein pairs, showing the source helices (a) and target helices colored by correspondences computed by **IterativeMatch** (b), **Match** (c), and the semi-rigid algorithm (d). Noise helices are shown with transparency, and unmatched target helices are colored black. Incorrectly matched target helices are outlined marked with the index of the source helix being wrongly matched with and the index of the correct source helix in parentheses.

24. J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, 23(1):31–42, Jan. 1976.
25. O. van Kaick, G. Hamarneh, H. Zhang, and P. Wighton. Contour correspondence via ant colony optimization. In M. Alexa, S. J. Gortler, and T. Ju, editors, *Pacific Conference on Computer Graphics and Applications*, pages 271–280, 2007.
26. L. Wang and U. Neumann. A robust approach for automatic registration of aerial images with untextured aerial lidar data. In *CVPR*, pages 2623–2630. IEEE, 2009.
27. Z. Yu and C. L. Bajaj. A structure tensor approach for 3d image skeletonization: Applications in protein secondary structure analysis. In *ICIP*, pages 2513–2516, 2006.
28. W. Zheng. Accurate flexible fitting of high-resolution protein structures into cryo-electron microscopy maps using coarse-grained pseudo-energy minimization. *Biophysical journal*, 100(2):478–488, 2011.
29. W. Zheng and B. R. Brooks. Normal-modes-based prediction of protein conformational changes guided by distance constraints. *Biophysical journal*, 88(5):3109–3117, 2005.
30. Y. Zheng and D. S. Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):643–649, 2006.