



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Line-Based object recognition using Hausdorff distance: from range images to molecular secondary structure

Concettina Guerra, Valerio Pascucci

December 13, 2004

Image and Vision Computing Journal

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Line-based object recognition using Hausdorff distance: from range images to molecular secondary structures

Concettina Guerra ¹ Valerio Pascucci ²

1. Università di Padova, Italy.

2. CASC, Lawrence Livermore National Laboratory, USA.

Keywords: 3D object recognition. Range images. Hausdorff distance. Geometric pattern matching. Molecular recognition.

Abstract

Object recognition algorithms are fundamental tools in automatic matching of geometric shapes within a background scene. Many approaches have been proposed in the past to solve the object recognition problem. Two of the key aspects that distinguish them in terms of their practical usability are: (i) the type of input model description and (ii) the comparison criteria used.

In this paper we introduce a novel scheme for 3D object recognition based on line segment representation of the input shapes and comparison using the Hausdorff distance. This choice of model representation provides the flexibility to apply the scheme in different application areas. We define several variants of the Hausdorff distance to compare the models within the framework of well defined metric spaces.

We present a matching algorithm that efficiently finds a pattern in a 3D scene. The algorithm approximates a minimization procedure of the Hausdorff distance. The output error due to the approximation is guaranteed to be within a known constant bound.

Practical results are presented for two classes of objects: (i) polyhedral shapes extracted from segmented range images and (ii) secondary structures of large molecules. In both cases the use of our

approximate algorithm allows to match correctly the pattern in the background while achieving the efficiency necessary for practical use of the scheme. In particular the performance is improved substantially with minor degradation of the quality of the matching.

1 Introduction

In this paper we present a method for comparing a model of a 3D object with a range image under rigid transformations. Both the model and the image object are represented in terms of line segments. The method is based on the computation of the Hausdorff distance between all the transformations of the set of model segments and the set of image segments. The Hausdorff distance is a max-min distance that has been often used in computer vision for object recognition, mostly applied to the case of 2D pointsets. The Hausdorff distance easily generalizes to sets of line segments either in 2D or 3D space. However, the classical definition does not seem adequate for comparing objects represented as sets of line segments. Here we present variants of the basic definition of the Hausdorff distance between sets of line segments that are more suitable for object recognition.

The computation of the Hausdorff distance does not necessarily produce a one-to-one correspondence between the elements of the two sets; it may happen, in fact, that multiple elements in one set are associated with a single element of the other set. This is unlike most existing object recognition methods that give an explicit pairing. Another feature of the Hausdorff distance is its sensitiveness to occlusion since it does not allow to compare sub-patterns. However, extensions of the definition may overcome this problem and make the comparison between subsets of points or segments possible [17].

Much work has been done on the computation of the Hausdorff distance. In the area of computational geometry [11, 12] exact algorithms have been studied for the problem of deciding whether there exist a transformation that maps one set of points into another set within a given distance. Fundamental robustness issues are discussed in [2]. Exact algorithms cannot be used in most practical applications where measurement errors and noise are present; furthermore, the high computational complexity of the exact algorithms make them impractical for use in real problems. For these reasons, approximate solutions for the case of pointsets, both in 2-dimensional and in 3-dimensional space, have been considered [20].

In the field of computer vision, an efficient multi-resolution technique for comparing images using the Hausdorff distance has been presented in [17] where the space of possible transformations is limited to translations and scaling; in [33] the above technique is extended to affine transformations. Affine transformations are used in [23] for matching pointsets. The problem of matching sets of segments in images using a multidimensional Hausdorff distance has recently been considered in [35], where the objects to be matched are 2-dimensional. Other approaches to matching sets of segments in 3D space based on various techniques and metrics are given in [3, 5, 10, 23, 26]

In this paper we introduce variants of the definition of the Hausdorff distance between sets of segments and present an approximate algorithm for their efficient computation. We show that the error introduced with the approximation is within a bounded factor from optimal. This bound is the same as the bound obtained in [20] for the simpler case of pointsets.

We have implemented the matching procedure and tested it on real range image data. A comparison of the proposed matching procedure with a more extensive search that examines a larger transformation space has been performed on several range images. The comparison confirms that the quality of the results of our practical approach is generally good (better than the theoretically guaranteed error bound). We also show how the scheme can be applied to other application fields like molecular pattern matching, so long as one can use the line-based model representation scheme.

The following of this paper is organized as follows. Section 2 reviews the definition of the Hausdorff distance and introduces different metrics for comparing sets of segments, that are suitable for computer vision applications. Section 3 shows properties of these metrics that allow fast computation of the distance itself. The matching procedure is presented and analyzed in section 4. Section 5 presents and compares experimental results both on range data obtained with different metrics and on secondary structures of molecular data obtained from the Protein Data Bank. It also contains a brief discussion of the problem of 3D line detection and of the procedure used in our approach to extract line segments from range images. In general the segmentation of a range image is in itself a fundamental problem [6, 24]. Obviously, the variations of the input generated by the different algorithms can have significant impact on the result generated by the matching procedure. An in depth discussion of the quality of the segmentation produced by each approach is far beyond the scope of this paper. A good survey of different techniques and a framework for evaluating the quality of a segmentation

scheme can be found in [24].

2 The Hausdorff Distance between Sets of Segments

We denote a set of straight line segments by an uppercase letter (A, B, \dots) , a single straight line segment by a lowercase letter (a, b, \dots) , the start and end point of segment a by a^s and a^e and a generic point by a Greek letter (α, β, \dots) . We compare two sets $A = \{a_1, a_2, \dots, a_m\}$ and $B = \{b_1, b_2, \dots, b_n\}$ of line segments a_i and b_j .

Def. 1 (*Hausdorff distance*)

The Hausdorff distance $H(A, B)$ between A and B is:

$$H(A, B) = \max(h(A, B), h(B, A)),$$

where $h(A, B)$ is the one-way Hausdorff distance from A to B given by:

$$h(A, B) = \max_{\alpha \in a_i \in A} \left(\min_{\beta \in b_j \in B} d(\alpha, \beta) \right),$$

and $d(\alpha, \beta)$ is the Euclidean distance between two points α and β .

In the following we will use, with some abuse of notation, the same symbol $H(\cdot, \cdot)$ for the Hausdorff distance between sets of points and sets of segments. For the case of points in the definition above $\alpha \in a_i \in A$ is replaced by $\alpha_i \in A$ and $\beta \in b_j \in B$ is replaced by $\beta_j \in B$.

The basic Hausdorff distance provides a good metric over pointsets but does not preserve the notion of relevant subsets like the segments (see [12]). In our application we wish to keep information relative to the identity of distinct line segments since they are the basic elements in the description of the objects. Thus we introduce the following definition:

Def. 2 (*Segment Hausdorff distance*)

The Segment Hausdorff distance $H_S(A, B)$ between A and B is:

$$H_S(A, B) = \max(h_S(A, B), h_S(B, A)),$$

where $h_S(A, B)$ is the one-way Segment Hausdorff distance given by:

$$h_S(A, B) = \max_{a_i \in A} \left(\min_{b_j \in B} H(\{a_i\}, \{b_j\}) \right)$$

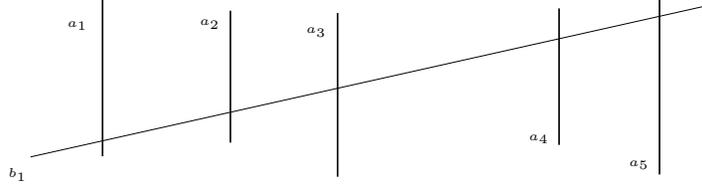


Figure 1: The standard Hausdorff metric function $H(\{a_1, \dots, a_5\}, \{b_1\})$ has a low distance value. The metric function $H_S(\{a_1, \dots, a_5\}, \{b_1\})$ has a higher value providing a more accurate similarity information.

Since $h_S(A, B) = \max_{a_i \in A} (\min_{b_j \in B} H(\{a_i\}, \{b_j\}))$ where $H(\{a_i\}, \{b_j\})$ is the Hausdorff distance (under definition 1) between two single segments, one can easily show that $H_S(A, B)$ defines a metric¹.

Consider the case of Figure 1. The standard Hausdorff metric function $H(\{a_1, \dots, a_5\}, \{b_1\})$ has a low distance value since each portion of the segment b_1 is near one of the segments a_i . On the contrary, the distance $H_S(\{a_1, \dots, a_5\}, \{b_1\})$ has higher value since there is no single segment a_i with low distance from b_1 . Thus the information provided by H_S is more accurate than the information provided by H .

To simplify the computation we introduce an equivalent, within a given error factor, simplified distance function.

Def. 3 (*Simplified Segment Hausdorff distance*)

The *Simplified Segment Hausdorff distance* $H_{SS}(A, B)$ between A and B is:

$$H_{SS}(A, B) = \max(h_{SS}(A, B), h_{SS}(B, A)),$$

where $h_{SS}(A, B)$ is the one-way *Simplified Segment Hausdorff distance* given by:

$$h_{SS}(A, B) = \max_{a_i \in A} \left(\min_{b_j \in B} H(\{a_i^s, a_i^e\}, \{b_j^s, b_j^e\}) \right) \quad (1)$$

$H_{SS}(A, B)$ is a metric. We show that it is equivalent to $H_S(A, B)$ in the sense that:

$$H_S(A, B) \leq H_{SS}(A, B) \leq \sqrt{2}H_S(A, B). \quad (2)$$

The advantage of this simplified distance function is not only the ease of computation but the fact that it can be also modified to take into account

¹Recall that a distance function $d(a, b)$ defines a metric if: (i) $d(a, b) \geq 0$; (ii) $d(a, b) = 0 \Leftrightarrow a \equiv b$; (iii) $d(a, b) = d(b, a)$; (iv) $d(a, b) \leq d(a, c) + d(c, b)$.

the orientations of the segments. All the distance functions defined above fail to carry this notion since:

$$H_{SS}(\{\overline{a^s a^e}\}, \{\overline{a^e a^s}\}) = H_S(\{\overline{a^s a^e}\}, \{\overline{a^e a^s}\}) = H(\{\overline{a^s a^e}\}, \{\overline{a^e a^s}\}) = 0$$

Def. 4 (*Oriented Segment Hausdorff distance*)

The Oriented Segment Hausdorff distance $H_{OS}(A, B)$ between A and B is:

$$H_{OS}(A, B) = \max(h_{OS}(A, B), h_{OS}(B, A)),$$

where $h_{OS}(A, B)$ is the one-way Simplified Segment Hausdorff distance given by (remember that a^s and a^e are the endpoints of a):

$$h_{OS}(A, B) = \max_{a_i \in A} \left(\min_{b_j \in B} \left(\max \left(d(a_i^s, b_j^s), d(a_i^e, b_j^e) \right) \right) \right) \quad (3)$$

It is easy to show that H_{OS} is a metric.

Def. 5 (*Minimal Segment Hausdorff distance*)

The Minimal Segment Hausdorff distance $H_{MS}(A, B)$ between A and B is:

$$H_{MS}(A, B) = \max(h_{MS}(A, B), h_{MS}(B, A)),$$

where $h_{MS}(A, B)$ is the one-way Minimal Segment Hausdorff distance given by:

$$h_{MS}(A, B) = \max_{a_i \in A} \left(\min_{b_j \in B} D(a_i, b_j) \right) \quad (4)$$

where $D(a_i, b_j)$ is the minimal distance between two segments defined as follows. Let r and l be the lines that each contain one segment. Assume that the two lines are non intersecting. Then the minimum distance d between r and l is given by:

$$d = \lambda(x_1 - x_2) + \mu(y_1 - y_2) + \nu(z_1 - z_2)$$

where $P_1 = (x_1, y_1, z_1)$ is a point of r , $P_2 = (x_2, y_2, z_2)$ is a point of l , and λ, μ, ν are the direction cosines of the perpendicular to both the given lines. If the lines intersect their minimal distance is zero. If the points separated by the distance d are within the two segments a_i and b_j , then $D(a_i, b_j) = d$; otherwise, the minimum distance will involve endpoints of one or both of the segments, as in the previous definition. Clearly, $H_{MS}(A, B) \leq H_{SS}(A, B)$.

$H_{MS}(A, B)$ is easy to compute, however it may not be a good metric for matching segments in computer vision, because, for instance, segments that differ significantly in orientation and length may have a very small minimal distance value.

All the above definitions are based on the computation of a max value in a given set of computed values. Often, in computer vision substituting the max function by the average of all the values in the set may lead to better results. In [14] 24 variations of the Hausdorff distance are compared for 2D pointsets in the presence of noise. Of all of them, the one based on the *average distance* between the points of one set to the other set is shown to give the best results for object recognition.

In our tests, we have experimented with all the above definitions using both the max and the average function to find the best matching.

3 Properties of the Segment Hausdorff distance

In this section we present a property of the distance function H_{SS} that is crucial to the computational efficiency of the method. We prove that using this norm, we can reduce the problem of nearest neighbor among segments to a query for a nearest neighbor among points. We first concentrate on the Oriented Distance $H_{OS}(A, B)$ then we show how to reformulate $H_{SS}(A, B)$ so that we can apply to it the same technique used for $H_{OS}(A, B)$.

Consider equation (3) and assume that A is a set of m segments and B is a set of n segments. The explicit computation of the maximization/minimization $\max_{a_i \in A} (\min_{b_j \in B} (\cdot))$ would require $O(mn)$ time complexity.

We consider an approximate solution that achieves a $O(m \log n)$ time complexity by reducing the problem to a nearest neighbor query among points in \mathbb{R}^6 under the mapping \mathcal{M} defined below. Consider the segment $a = \overline{a_s, a_e}$, with $a_s = \{x_s, y_s, z_s\}$ and $a_e = \{x_e, y_e, z_e\}$, then $\mathcal{M}(a)$ is the point $\{x_s, y_s, z_s, x_e, y_e, z_e\}$. Formally:

$$\mathcal{M}(\mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^6) :$$

$$(\{x_s, y_s, z_s\}, \{x_e, y_e, z_e\}) \mapsto \{x_s, y_s, z_s, x_e, y_e, z_e\}$$

The following theorem states that $\max(d(a_i^s, b_j^s), d(a_i^e, b_j^e))$ is a Minkowski metric in the range of the mapping \mathcal{M} .

Theorem 1 Consider a Minkowski norm $\|\cdot\|_M$ in \mathfrak{R}^3 . The function $\|\cdot\|_M^*$ defined as:

$$\begin{aligned} \|\{x_1, x_2, x_3, x_4, x_5, x_6\}\|_M^* &= \\ &= \max(\|\{x_1, x_2, x_3\}\|_M, \|\{x_4, x_5, x_6\}\|_M) \end{aligned} \quad (5)$$

is a Minkowski norm in \mathfrak{R}^6 .

Proof: To prove the theorem we just need to show that the unit ball $B^* = \{X : \|X\|_M^* \leq 1\}$ is a symmetric convex set (see Proposition 1.1.8 in [34]).

From equation (5) it follows that the unit ball B^* is the pointwise Cartesian product of the two unit balls B_1 and B_2 defined by the norm $\|\cdot\|_M$ and properly embedded in the coordinate subspaces (x_1, x_2, x_3) and (x_4, x_5, x_6) . That is iff $(x_1, x_2, x_3) \in B_1$ and $(x_4, x_5, x_6) \in B_2$ then $(x_1, x_2, x_3, x_4, x_5, x_6) \in B^*$.

Now observe that, being $\|\cdot\|_M$ a Minkowski norm, both B_1 and B_2 are symmetric convex sets. It follows that:

$$B^* = B_1 \times B_2$$

is in turn a symmetric convex set. Hence $\|\cdot\|_M^*$ is a Minkowski norm. \diamond

Theorem 1 allows us rewrite the *min* term of expression (3) as:

$$\min_{b_j \in B} (\|b - a\|_M^*).$$

This expression can be evaluated in optimal $O(\log n)$, within an approximation factor of $1 + \epsilon$, using the search technique presented in [4]. It follows that $h_{OS}(A, B)$ can be determined in $O(m \log n)$ (after a preprocessing of $O(n \log n)$). $H_{OS}(A, B)$ can be computed within the same $1 + \epsilon$ approximation factor in time $O(m \log n + n \log m)$ (after a preprocessing of $O(m \log m + n \log n)$).

To extend the above result to the case of Simplified Distance $H_{SS}(A, B)$ we introduce a reduced formula for its computation.

Theorem 2 The Simplified Segment Hausdorff distance between two segments is given by the expression:

$$H_{SS}(\{a\}, \{b\}) = \min(\max(d(a_s, b_s), d(a_e, b_e)), \max(d(a_s, b_e), d(a_e, b_s))) \quad (6)$$

Proof: From Definition 3 we have:

$$\begin{aligned}
H_{SS}(\{a\}, \{b\}) &= \max(\max(\min(d(a_s, b_s), d(a_s, b_e)), \min(d(a_e, b_s), d(a_e, b_e))), \\
&\quad \max(\min(d(b_s, a_s), d(b_s, a_e)), \min(d(b_e, a_s), d(b_e, a_e)))) \\
&= \max(\min(d(a_s, b_s), d(a_s, b_e)), \min(d(a_e, b_s), d(a_e, b_e)), \\
&\quad \min(d(b_s, a_s), d(b_s, a_e)), \min(d(b_e, a_s), d(b_e, a_e))) \quad (7)
\end{aligned}$$

Among all the 24 permutation of the $d(a_i, b_j), i, j \in s, e$ we only need to consider three groups the remaining being obtained by symmetry.

Observe that being $d(a_s, b_s) = d(b_s, a_s), d(a_s, b_e) = d(b_e, a_s), d(a_e, b_s) = d(b_s, a_e)$ and $d(a_e, b_e) = d(b_e, a_e)$, the $\min(\cdot, \cdot)$ comparisons in expression (7) involve only four values. Call the four distinct distance values $\mu_1, \mu_2, \mu_3, \mu_4$ and assume that $\mu_1 \geq \mu_2 \geq \mu_3 \geq \mu_4$. In the expression (7) only four $\min(\mu_i, \mu_j)$ pairwise comparisons are considered out of all the six non-symmetric ones. Only the remaining two comparisons are present in equation (6) as $\max(\mu_i, \mu_j)$ instead of $\min(\mu_i, \mu_j)$. Furthermore, if μ is present in (7) then it appears twice in it.

Observe also that $H_{SS}(\{a\}, \{b\})$ is either μ_2 or μ_3 because in expression (7) one of the three comparisons $\min(\mu_1, \mu_2), \min(\mu_2, \mu_3)$ and $\min(\mu_1, \mu_3)$ must be present.

In particular if expression (7) contains $\min(\mu_1, \mu_2) = \mu_2$ then $H_{SS}(a, b) = \mu_2$ because the outer $\max(\cdot, \cdot, \cdot, \cdot)$ comparison will give as result μ_2 . In this case expression (6) will not contain the $\max(\mu_1, \mu_2)$ but will instead contain either $\max(\mu_2, \mu_3) = \mu_2$ or $\max(\mu_2, \mu_4) = \mu_2$. In both cases the outer $\min(\cdot, \cdot)$ of expression (6) will return the value μ_2 like expression (7). Otherwise, if expression (7) does not contain $\min(\mu_1, \mu_2)$ then $H_{SS}(a, b) = \mu_3$. because either the comparison $\min(\mu_1, \mu_3) = \mu_3$ or the comparison $\min(\mu_2, \mu_3) = \mu_3$ must be present. In this case expression (6) must involve $\max(\mu_1, \mu_2) = \mu_1$. The second comparison will be $\max(\mu_3, \mu_4) = \mu_3$ thus the outer $\min(\cdot, \cdot)$ comparison returns μ_3 .

In conclusion expressions (7) and (6) have the same value. \diamond

Plugging equation (6) into (1) allows us to compute H_{SS} with exactly the same technique used to compute H_{OS} . The only difference is that expression (6) replaces the norm $\max(d(a_i^s, b_j^s), d(a_i^e, b_j^e))$. This can be easily fixed by observing that the outer $\min(\cdot)$ of the (6) will be evaluated by the nearest neighbor query. Hence we just need to modify the map \mathcal{M} so that

the segment b is not mapped only to the point $\{x_s, y_s, z_s, x_e, y_e, z_e\}$ but also to the point $\{x_e, y_e, z_e, x_s, y_s, z_s\}$. Hence we apply the same $O(\log n)$ search algorithm but on a set of points in \mathbb{R}^6 of cardinality double the cardinality of B . The overall time complexity for the computation of $H_{SS}(A, B)$ is $O(n \log m + m \log n)$.

4 The Matching algorithm

In this section we describe an algorithm for approximate matching of sets of segments under rigid body transformations. The problem can be formulated as follows. Given two sets of line segments, the pattern set A and the model set B , and given a similarity measure d , find the rigid transformation g that minimizes the distance $d(g(A), B)$.

The algorithm proposed has been applied with the metrics H_S , H_{SS} and H_{OS} between sets of segments. It is based on ideas first developed in [20] for pointset pattern matching here extended to the case of sets of segments. Conceptually the algorithm can be divided in three main stages:

- determine a translation t ;
- determine a rotation r ;
- evaluate the distance between $g(A)$ and B , where g is the combined transformation.

For the determination of an appropriate translation and rotation we select three “representatives” segments for each of the two sets A and B . The representatives must be affinely independent.

In the first step we randomly pick one representative segment a for A . Then we choose one representative b for B . Since, for any matching, a must be paired with an element of B then we keep a fixed and try all its possible pairings with the n elements of B . The translation t is defined by taking the mid-point a_m of a into the mid-point b_m of b . This choice of the translation minimizes the distance between the transformed segment $t(a)$ and b for any fixed orientation. In many existing pattern matching algorithms, the translation component of the geometric transformation is computed separately by aligning the centroids of the two patterns. This reduces the degrees of freedom thus making the entire computation of the transform more efficient. There are however often problems associated with

this heuristic especially when large portions of an object may be occluded. The method we use here, even though more computationally intensive, is more reliable in presence of occlusion.

To define the rotation we need two additional independent elements of qA . They are selected so that the error due to the approximation is maintained within a guaranteed bound. In particular the second representative a' is the segment containing the point a'_f farthest from a_m . The third representative a'' contains the point a''_d at maximum distance from the line $\overline{a_m a'_f}$. It is easy to see that the points a'_f and a''_d must each be an endpoint of some segment. Note that the segment a'' does not need to be distinct from the segments a and a' . The condition that we enforce is instead the affine independence of the three points a_m , a'_f and a''_d .

The next step of the algorithm is to choose the segments b' and b'' of B in all the m^2 possible ways. For each b' and each endpoint of b' , consider the rotation that has origin in a_m and that makes a'_f and a_m to become collinear with the endpoint of b' . Define r' as the one of the above rotations that minimizes the distance between a'_f and an endpoint of b' . Then define r'' to be the rotation about the axis $a_m a'_f$ that brings a''_d closest to an endpoint of b'' . Apply the transformations $r''(r'(t(A)))$. Finally choose over all the triplets b , b' and b'' the combined transformation g that resulted in the smallest distance.

We show the following:

Theorem 3 *The proposed matching algorithm generates a rigid transformation that results in a directed H_{SS} distance that is at most eight times larger than the optimal.*

Proof: Let g_o be the optimal rigid transformation, i.e. the one that minimizes the H_{SS} distance between the transformed pattern and the model. Furthermore, let H_{SS}^o its corresponding optimal distance value. For each segment a of $g_o(A)$ there is a corresponding segment of B that is within H_{SS}^o distance from a . Translate the pattern $g_o(A)$ so that the midpoint a_m of a becomes coincident with the midpoint of its corresponding segment in B , call it b_m . Since

$$d(a_m, b_m) \leq \min(\max\{d(a^s, b^s), d(a^e, b^e)\}, \max\{d(a^s, b^e), d(a^e, b^s)\}) = H_{SS}(\{a\}, \{b\}) \leq H_{SS}^o \quad (8)$$

every other segment of the pattern is moved by this translation by a distance of at most H_{SS}^o . Consider now a'_f and the closest endpoint of its correspond-

ing segment of B , say b'_e . a'_f and b'_e are at a distance at most H_{SS}^o . Thus the rotation with origin in a_m that causes a_m , a'_f and b'_e to be collinear, moves a'_f by at most $2H_{SS}^o$. Since any other point of the pattern is closer to a_m than a'_f , it will be moved by a smaller amount. Finally, consider the segment b'' of B corresponding to a'' . The rotation about the axis $a_m a'_f$ that brings a''_d closest to an endpoint of b'' moves any other point of the pattern by at most $4H_{SS}^o$. In conclusion, after applying all three transformations any point of A will be moved by at most $7H_{SS}^o$ and therefore will be at a distance from its closest point in B of at most $8H_{SS}^o$. Since our algorithm will have considered the above transformation among all others, it would have generated a solution that is at most 8 times worse than the best one. \diamond

If we take into account the error factor ϵ due to the approximation of the nearest neighbor computation, we obtain an error factor of 8ϵ . Similarly, for the H_S we have the following:

Theorem 4 *The proposed matching algorithm based on the H_S metric generates a rigid transformation that results in a directed H_S distance that is a factor of $(7 + \sqrt{2})$ larger than the optimal.*

Proof: The proof is similar to the previous one and uses the relation (2). It is omitted. \diamond

The time complexity of the algorithm is $O(mn^3 \text{NearestSegment}(n))$, assuming m segments in the image and n segments in the model, where $\text{NearestSegment}(n)$ is the time to determine the closest segment in a set of n segments.

As we have seen in the previous section, using the Hausdorff metric H_{SS} (or H_{OS}) the nearest neighbor query in a set of segments (to identify the segment of the model segment “closest” to a segment of the image), reduces to a nearest neighbor query among points in \mathcal{R}^6 that can be performed in optimal $O(\log n)$ time within a known error bound. Then, the overall worst case time complexity for the matching algorithm with the H_{SS} distance is $O(mn^3 \log n)$.

Note that, to the best of our knowledge, no algorithm is known for the exact computation of the Hausdorff distance in 3D space under rigid transformations. For a review and analysis of the exact Hausdorff distance computation in 2D for rigid transformations and in 3D, for translations only, see [20].

5 Experimental results

The program PM written in C++ that implements the proposed matching strategy can use various definitions of distance between sets of segments. The inputs to PM are two lists of segments A and B , the output consists of a list of corresponding segments of the two input sets, a distance value for each corresponding pair and a global distance value. This latter value may be the maximum over all distance values, as in the standard Hausdorff distance, or the average.

Here we report on the results obtained using the one-way segment Hausdorff distance h_{SS} on two types of input data: range images and protein secondary structures.

5.1 Range images

Line and Segment extraction

We first shortly describe how the segments are extracted from the images. We have implemented two strategies to extract line segments from range data [22]. The strategies generate the set of 3-dimensional lines that best fit the data based on *random sampling*. Methods that are very common for line detection in 2D, like the Hough transform [15, 30], have been not considered because of their high memory requirements when extended to 3D. The two strategies are applied to the set of edge points detected in a range image using the scanline approximation approach of [25]. We have also used as input to the line extraction procedure, to be described below, the set of boundary points derived from range image segmented into planar regions [6, 24].

The simplest line detection approach finds the best line among the lines defined by pairs of input edge points. For each such line, it counts the number of edge points within ϵ distance from it, where ϵ is a given tolerance threshold; the best line is the one that produces the maximum value. RANSAC [7] restricts the search for the best line to a random selected subset of pairs of edge points, thus keeping the computational complexity of the approach reasonably low. For the detection of multiple lines, the above procedure is applied repeatedly after the removal from the set of edge points of all points within $2 * \epsilon$ from the best line.

The second strategy we have implemented is a robust optimization technique based on *tabu search* that explores a larger set of candidate lines. Tabu

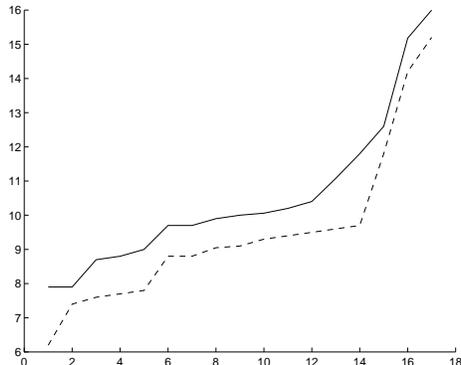


Figure 2: Comparison of the two matching strategies. The distance values are represented on the y axis for 18 datasets. The solid line gives the distance values obtained by our fast algorithm; the dashed line by the exhaustive search. The 18 experiments are sorted by increasing value of the distances obtained by our fast algorithm.

search (TS) [19] is a powerful optimization technique that has been used to solve a variety of complex combinatorial problems. One of the main components of TS is the use of adaptive memory: during the search, local choices are guided by the past history of the search. Restrictions are imposed by making reference to the memory structures storing the tabu or forbidden alternatives. This prevents solutions from the recent past from being revisited. We have applied the basic TS paradigm to the line detection problem and obtained results of better quality than using the above simple approach. The quality of a line is defined as number of edge points within ϵ distance from the line. The method does not restrict the lines to pass through pairs of edge points and therefore may find a better fit of lines to points especially for long line segments.

Matching

The inputs to PM are two lists of segments A and B of two images or of an image and a model. The output of the algorithm also includes the rigid transformation (rotation and translation) that maps one set of segments into the other.

To evaluate our matching method we have compared it against a more extensive search that examines a larger transformation space. This second method examines all triples of segments of the set A and finds the best

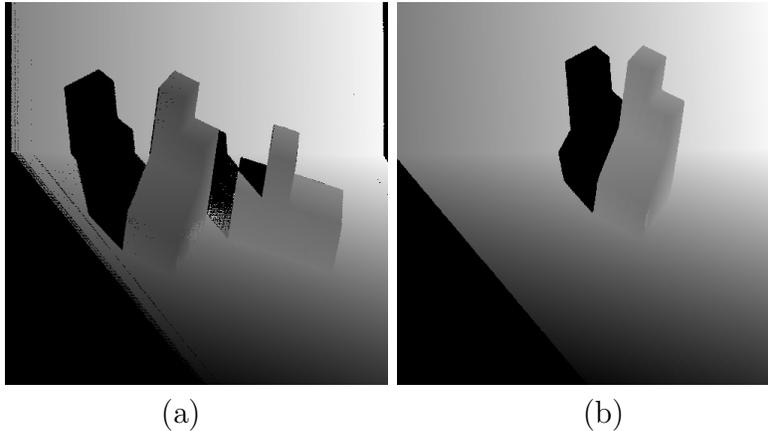


Figure 3: The range images `abw.train.3` (a) and `abw.test.27` (b)

transformation among those obtained by associating each triple of A with all triples of segments of B . The time complexity of this more extensive search is very high ($O(n^6 \log n)$) and unacceptable in practice. There is no guarantee that this new method produces the optimal Hausdorff distance; however, it may be closer to that value. A comparison of the distance values obtained by the two search methods applied to several pairs of images is displayed in Figure 2. The distance values in the figure represent max values as in the standard Hausdorff definition. The figure shows that the distances computed by the more extensive search are only marginally better.

We have run the program on several pairs of range images of polyhedral objects acquired by an ABW structures light scanner.² In our experiments we have computed the average Hausdorff value because it is less sensitive to errors due to spurious elements and thus generally gives more reliable results in model-based object recognition. This is in accordance with other experiments based on the Hausdorff distance for pointsets in computer vision [14], [35]. We report here on the results on a pair of images for which the line segmentation process generates several line segments that do not correspond to actual edges of the images and are due to noise. This makes the matching process more difficult. Consider the two images `abw.test.27` and `abw.train.3` shown in Fig. 3, and match the first against the second image. The segments are extracted from the two range images by the Tabu Search. Figure 4 shows the segments of the two sets after the computed transformation is

²The images are available from <http://marathon.csee.usf.edu/range/segcomp/SegComp.html>.

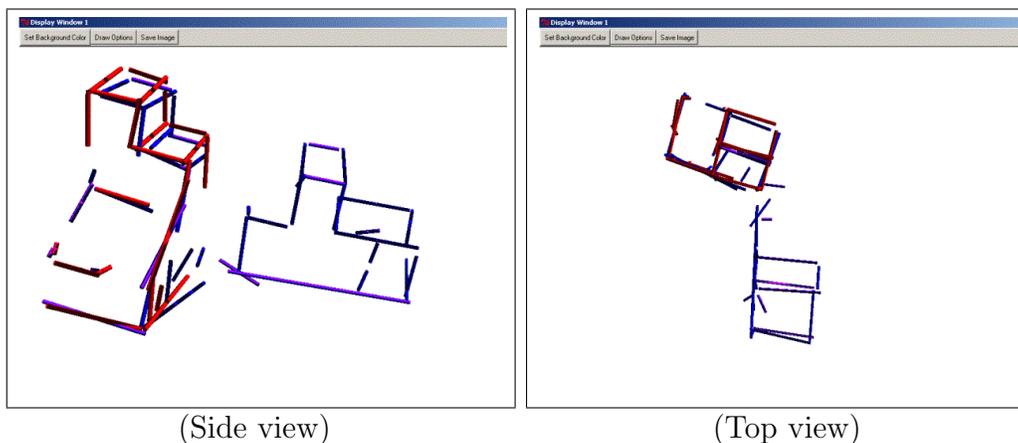


Figure 4: Two different views of the sets of segments of `abw.train.3` (black) and `abw.test.27` (red) after the computed transformation is applied to the segments of `abw.test.27`

applied to the segments of `abw.test.27`. Despite the poor quality of the image segmentation, the output of the matching procedure is good. The obtained distance value (average) is 8.4. The rotational matrix is :

$$\begin{vmatrix} 0.99 & -0.06 & 0.05 \\ 0.07 & 0.99 & -0.07 \\ -0.05 & 0.07 & 0.99 \end{vmatrix}$$

while the parameters of the translation are: $[-66.64, -7.45, 57.38]$.

Figure 5 shows the results obtained when matching a portion of a cube (black lines) with three different polyhedral objects (grey lines).

The execution time for the range images of figures 3-4 was approx 5 minutes. For very large datasets, clever pruning strategies may reduce the computation time by trying to eliminate candidate associations from further examination early in the process.

5.2 Protein secondary structure

PM was used for pairwise comparison of protein structures and tested with many different proteins from the Protein Data Bank (PDB) [1]. The protein structural comparison has received a lot of attention in recent years for its relevance in bioinformatics and functional genomics. There are many reasons

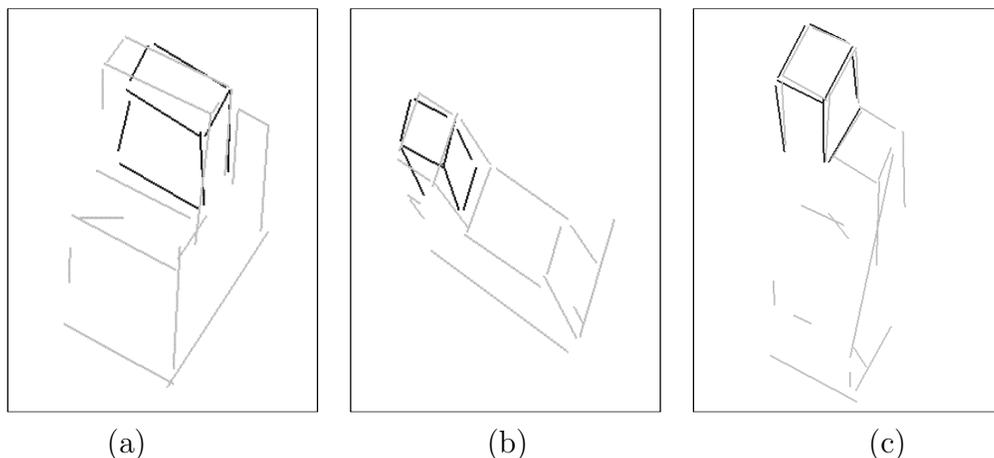


Figure 5: (a-c) Matching of a simple cube pattern (black lines) with three different backgrounds of polyhedral objects (gray lines).

why protein structure comparison is important. First, since the structure of a protein is intrinsically related to its function, structural comparison can help to assign a function to a newly determined protein structure from its similarity to a protein with known function. Second, it can be used for protein classification to build a library of 3D shapes of proteins or folds. Third, it is a valuable tool for evaluating and assessing the quality of various methods for proteins structure prediction.

The protein structure comparison may involve different levels of representations of the three dimensional structures, from the atomic level to the level of secondary structures. The *secondary structure* of a protein is a representation in terms of recurrent regular substructures, the α -helices and the β -strands, that play an important role in the functional behavior of a protein. Arrangements of the α -helices and β -strands are the basis for the protein structural classification of SCOP [29]. For a survey of the protein architecture see [8], [28].

Most methods presented in the literature for fold comparison deal with a protein representation in terms of atomic coordinates. Approaches based on secondary structures have been used mostly for fast retrieval of folds or motifs from the PDB. Furthermore, the comparison of secondary structures can be used as the first step in a comparison procedure that first identifies possible candidate solutions in a fast way and then refines the solutions taking

into consideration the more detailed atomic descriptions of proteins [13]. Extensive surveys on the subject of protein comparison exist focusing on different aspects of the problem [9], [16], [27].

We have used our approach to segment matching to compare proteins at the level of secondary structural elements, with the α -helices and β -strands represented as linear segments. The segment associated to a β -strand is the best fit line segment for the set of atoms comprising the strand. For an α -helix, the associated segment is its geometric axis. One major distinction among the comparison approaches is whether they take into account the order of the secondary structure elements along the protein chain as a constraint in the correspondence process. Our matching procedure is order independent. Moreover, it allows to associate only secondary structures of the same type, i.e. helices to helices and strands to strands.

We have conducted experiments on several sets of segments of proteins structures. The test proteins were of different fold classes, either all α -structures, all β -structures, or α - β -structures combined. As already mentioned, the program PM provides as output a list of pairs of corresponding segments and for each such pair a distance value. In addition, it outputs the average (or the maximum) distance value over all pairs of segments and the obtained rigid transformation.

The line segments associated to secondary structures were determined by a line fitting procedure using either tabu search or a well-known technique based on singular value decomposition [18]. Only C_α atoms of the protein backbone were considered for the determination of the best fit segment. For both procedures the outputs were very satisfactory. There were very few exceptions corresponding to strands with a significant bent at the extremes. The examples shown in this section have been obtained with tabu search.

The first set of tests dealt with pairwise comparison of whole proteins. For proteins that are known to be structurally similar, the program successfully reported a large number of corresponding pairs of segments with small distance values. Consider, for instance, the two proteins 1rpa and 1rpt (in Figure 6(a-b)) that have very similar segments with only a small perturbation. Each protein consists of 13 helices and 14 strands. The superposition of the two sets of segments obtained by applying to one of the two proteins the obtained rigid transformation can be seen in Figure 6(c). The maximum Hausdorff distance value h_{SS} returned by our practical method was 1.22\AA while the average Hausdorff value was 0.7\AA . The more extensive search described in the previous section obtained a maximum Hausdorff value of 0.98

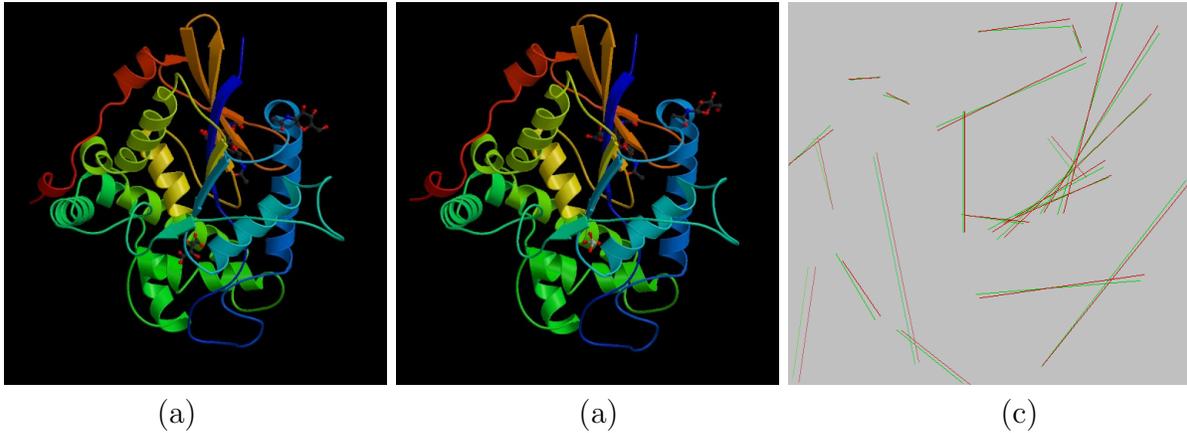


Figure 6: Segment based comparison of secondary structures. The secondary structure of the input proteins 1rpa (a) and 1rpt (b). (c) Fitting of the segments representing the two proteins (1rpa in green and 1rpt in red). The fitting is computed by minimizing the distance H_{SS} .

\AA thus only slightly better.

The average value compares well with the results of the server PROuST [13] (<http://angela.dei.unipd.it/PROuST/>) that gives an RMSD (Root Mean Square Deviation) of 0.4\AA between the corresponding atoms. PROuST uses a more complex procedure that takes into account both the secondary structures and the atomic coordinates; it is based on a combination of indexing techniques and dynamic programming.

A second set of experiments involved searching for protein substructures, i.e. specific geometric arrangements of secondary structures or motifs in sets of proteins. The β -barrel is a motif in which the secondary structures are arranged to form the shape of a torus [32]. The experiment used the β -barrel of some given protein, for instance the taka-amylase (PDB code:2taa), to define the motif and then matched it against a set of proteins to determine the ones containing a similar barrel. To validate our approach, the chosen motif was sought in a set of proteins some of which are known from the PDB to contain such a motif, as, for instance, hydrolase (2amg), triose phosphate isomerase (1tim), carbon-oxygen lyase (3enl), pseudoazurins (1paz), etc. The set was enlarged to include proteins without a barrel motif and with different folds structure, among which the azurins (2aza, 1azu), plastocyanins (6pcy), etc.

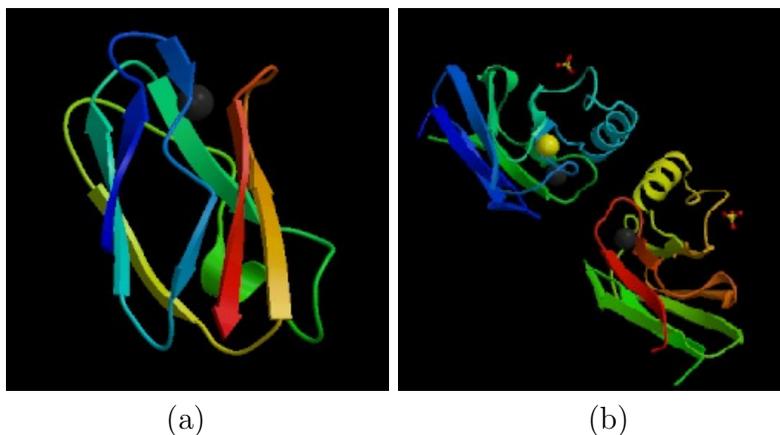


Figure 7: The secondary structure of proteins 6pca (a) and 2aza (b) for which the best fitting line segments are determined.

Barrels present in proteins may differ both in the number of strands, their length, and in general in their spatial arrangement. The β -barrel of 2taa chosen as search pattern comprises 8 β -strands. The protein 2taa contains a total of 9 α -helices and 19 β -strands. When matching the 8-stranded barrel with each protein of the set, good similarity results were obtained only for proteins known to contain a barrel. In such cases, the number of corresponding pairs obtained within a given small distance value was a considerably large fraction of the structures present in the motif. As an example, consider protein hydrolase 2amg that contains a total of 13 β -strands 8 of which forming a barrel. The pairwise comparison between the search pattern of 2taa and all secondary structures of 2amg resulted in 7 pairs of corresponding segments with distance less than 7.4\AA and an average Hausdorff value of 4.9\AA . The 7 segments of 2amg are indeed those of the barrel.

Another example used as search pattern the β -sheets of the plastocyanin structure 6pcy (in Figure 7) that are arranged to form a sandwich motif. Figure 8 shows the match between the pattern of 6pcy and protein azurin 2aza that also contains a sandwich motif. The match resulted in 8 strands being associated within a distance less than 8.1\AA and an average Hausdorff distance of 5.5\AA .

When the entire structure of 6pcy, which also includes a short helix, was compared with 2aza 9 pairs of corresponding segments were reported with

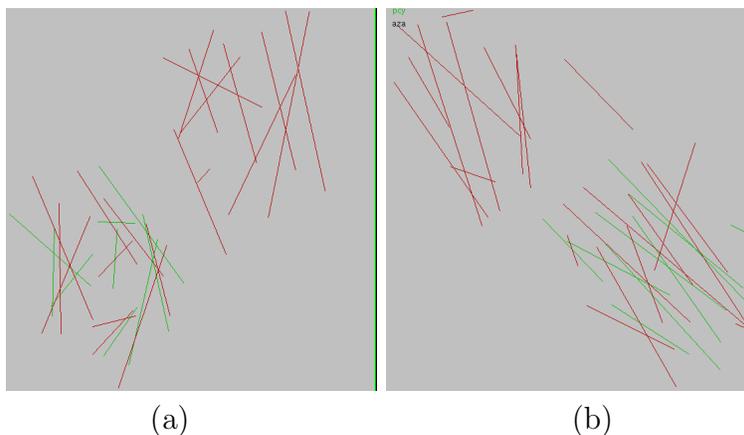


Figure 8: The segments of the two sets 6pcy and 2aza, with different colors, are shown after the computed transformation is applied to the segments of one set. (a) Top view. (b) side view.

distance less than 10.5 \AA

For all such comparisons the h_{SS} metric was used.

The above tests performed on proteins of known families showed that PM was successful in locating matches of proteins sub-structures common to members of a protein family.

The execution times for matching pairs of proteins are generally very good. The average number of secondary structures in the roughly 27,000 proteins now present in Protein Data Bank is 13. For the examples presented in this paper the execution time was of the order of seconds on a SUN Sparc 5 station.

6 Conclusions

In this paper, a method for matching 3D objects based on line segments and on the Hausdorff distance and its many variants was presented. The proposed method determines the rigid transformation that aligns one set of segments with the other. We have conducted experiments on different datasets: polyhedral objects extracted from range images and protein structures. For the range data, the preliminary line detection phase is a crucial and difficult step especially in the presence of noise and occlusion. To improve the performance of the method we have used tabu search for line detection and

obtained generally good results. The matching algorithm is however robust in the presence of errors in the segmentation process. To match protein structures, we have represented them in terms of secondary structures, i.e. helices and strands. The proposed strategy was particularly effective in searching for substructures in model databases.

7 Acknowledgements

Support for Guerra was provided in part by the Italian Ministry of University and Research under the FIRB Project "Bioinformatics for Genomics and Proteomics". The work of Pascucci was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48.

References

- [1] Abola, E. E., Sussman, J. L., Prilusky, J. and Manning, N. O. (1997). Protein Data Bank archives of three-dimensional macromolecular structures. *Methods Enzymol.*, 277, 556-571.
- [2] ALTER, T. D., GRIMSON, W.E.L. Fast and efficient 3D recognition by alignment. *ICCV93*, (1993), 113-120.
- [3] ARMAN, F., AGGARWAL, J.K. Model-based object recognition in dense-range images- A review. *ACM Computing Surveys*, (1993), 25(1), pp. 6-43.
- [4] ARYA, S., MOUNT, D. M., NETANYAHU, N. S., SILVERMAN, R., AND WU, A. An optimal algorithm for approximate nearest neighbor searching. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms* (1994), pp. 573-582.
- [5] BOYTER, B, AGGARWAL, J. K. Recognition of polyhedra from range data. *IEEE EXPERT*, 10 (1986), pp. 47-59.
- [6] BOCK, M. E., AND GUERRA, C. A geometric approach to the segmentation of range images. In *Proceedings of the Second International Conference on 3D Digital Imaging and Modeling* (Canada, October 1999), pp. 261-269.
- [7] BOLLES, C., AND FISCHLER, M. K. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24, 6 (1981), 381-395.

- [8] Branden, C., Tooze, J. (1999). *Introduction to Protein Structure*, (second edition), Garland.
- [9] Bourne P.E. and Weissig (Eds.) (2003). *Structural Bioinformatics* Wiley-Liss.
- [10] CHEN, H. H., AND HUANG, T. S. Matching 3d line segments with application to multiple-objects motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1990), 1002–1008.
- [11] CHEW, L. P., GOODRICH, M. T., HUTTENLOCHER, D. P., KEDEM, K., KLEINBERG, J. M., AND KRAVETS, D. Geometric pattern matching under Euclidean motion. *Comput. Geom.* 7, 1-2 (1997), 113–124. Fifth Canadian Conference on Computational Geometry (Waterloo, ON, 1993).
- [12] CHEW, P. L., DOR, D., EFRAT, A., AND KEDEM, K. Geometric pattern matching in d -dimensional space. In *Algorithms—ESA '95 (Corfu)* (Berlin, 1995), vol. 979 of *Lecture Notes in Comput. Sci.*, Springer, pp. 264–279.
- [13] COMIN, M., GUERRA, C., ZANOTTI, G. PROuST: a Comparison Method of Three-Dimensional Structures of Proteins using Indexing Techniques *Journal of Computational Biology*, Dec., 2004.
- [14] DUBUISSON, M., AND JAIN, A. A modified Hausdorff distance for object matching. In *Proc. Int. Conf. Pattern Recognition* (1994), pp. 566–568.
- [15] DUDA, R. O., AND HART, P. E. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* 15, 1 (Jan. 1972), 11–15.
- [16] Ferrari C., Guerra C. (2003). Geometric methods for protein structure comparison. *Protein Structure Analysis and Design*, (C. Guerra, S. Istrail, eds.), Lecture Notes in Bioinformatics, Springer-Verlag. 57-82.
- [17] HUTTENLOCHER, D. P., KLANDERMAN, G. A., AND RUCKLIDGE, W. J. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 9 (Sept. 1993), pp. 850-863.
- [18] GERSTEIN, M. A Resolution-Sensitive Procedure for Comparing Protein Surfaces and its Application to the Comparison of Antigen-Combining Sites. *Acta Cryst.*, A8, (1992), 271-276.
- [19] GLOVER, F., AND LAGUNA, M. *Tabu search*. Kluwer Academic Publishers, Boston, 1997.

- [20] GOODRICH, M. T., MITCHELL, J. S. B., AND ORLETSKY, M. W. Practical methods for approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (April 1999), pp. 371-380.
- [21] GRINDLEY HM, ARTYMIUK PJ, RICE DW, WILLETT P. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. *Journal of Molecular Biology*, 229, 3, (1993), 707-721.
- [22] GUERRA, C., PASCUCCI, V. Finding line segments in range images. *IEICE Transactions.* , E84-D, 12, (2001), 1739-1744.
- [23] HAGEDOORN, M., VELTKAMP, R. C. Reliable and efficient pattern matching using affine invariant metric. *Int. J. of Computer Vision*, 31, (2/3), (1999), pp. 203-225.
- [24] HOOVER, A., JEAN-BAPTISTE, G., JIANG, X., FLYNN, P. J., BUNKE, H., GOLDFOG, D., BOWYER, K., EGGERT, D., FITZGIBBON, A., AND FISHER, R. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (July 1996), pp. 1-17.
- [25] JIANG, X. Y., AND BUNKE, H. Edge detection in range images based on scan line approximation *Computer Vision and Image Understanding*, 73, 2, (1999), 183-199.
- [26] KANMGAR-PARSI, B., AND KAMGAMR-PARSI, B. Matching sets of 3d line segments with application to polygonal arc matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 10 (1997), pp. 1090-1099.
- [27] Lemmen, C., Lengauer, T. (2000). Computational methods for the structural alignment of molecules. *J. of Computer-Aided Molecular Design*, 14, 215-232.
- [28] Lesk, A. (1994). Computational Molecular Biology. *Encyclopedia of Computer Science and Technology*, 31, Marcel Dekker, Inc..
- [29] Murzin, A.G., Brenner, S.E., Hubbard, T., Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Molecular Biology*, 247, 536-540.
- [30] PRINCEN, J., ILLINGWORTH, J., AND KITTLER, J. A hierarchical approach to line extraction based on the Hough transform. *Computer Vision, Graphics, and Image Processing* 52, 1 (Oct. 1990), pp. 57-77.

- [31] REQUICHA, A. A. G. Representation for rigid solids: Theory, methods, and systems. *Computing Surveys* 12, 4 (1980), pp. 437-464.
- [32] RICHARDSON, J.S. β -sheet topology and the relatedness of proteins. *Nature*, 268, pp. 495–500, 1977.
- [33] RUCKLIDGE, W. J. Efficiently locating objects using the Hausdorff distance. *International Journal of Computer Vision* 24, 3 (1997), pp. 251-270.
- [34] THOMPSON, A. C. *Minkowski Geometry*, vol. 63 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1996.
- [35] YI, X., CAMPS, O. I. Line-based recognition using a multidimensional Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 9 (Sept. 1999), pp. 901-916.