Partial Surface and Volume Matching in Three Dimensions

Gill Barequet and Micha Sharir

Abstract—In this paper we present a new technique for partial surface and volume matching of images in three dimensions. In this problem we are given two objects in 3-space, each represented as a set of points, and the goal is to find a rigid motion of one object which makes a sufficiently large portion of its boundary lying sufficiently close to a corresponding portion of the boundary of the second object. This is an important problem in pattern recognition and in computer vision, with many industrial, medical, and chemical applications. Our method treats separately the rotation and the translation components of the Euclidean motion that we seek. The algorithm steps through a sequence of rotations, in a steepest-descent style, and uses a novel technique for scoring the match for any fixed rotation. Experimental results on various examples, involving data from industrial applications, medical imaging, and molecular biology, are presented, and show the accurate and robust performance of our algorithm.

Index Terms—Geometric hashing, computer vision, pattern recognition, partial surface matching, protein matching, molecule docking.

1 Introduction

THE problem of finding a full or a partial match between three-dimensional objects attracted considerable attention in the literature during the past decade. A main motivation comes from the object recognition problem in computer vision, where an object is viewed by a range sensor, and the resulting image has to be matched against a library of model objects. The image may contain several of the model objects (as well as other objects), and these objects may be only partially visible because of occlusion and because the sensor usually cannot scan all sides of the objects. In addition, the image is likely to be very noisy. The goal is to find a Euclidean motion of a model object, which makes it overlap a large portion of some object in the image, leading to the identification of the model object in the viewed scene, and to finding its position and orientation there. This is acknowledged by many researchers as a major problem in object recognition (e.g., [11, p. 137]). It has important applications for robot task planning, assembly, inspection, and many additional industrial, military, and other applications. Another significant motivation for the surface matching problem is docking of proteins in molecular biology, where a geometric fit between parts of the boundaries of two molecules (i.e., a partial surface matching) is sought, requiring also that the molecules do not overlap near the matched boundaries. Important applications of molecule docking are the recognition and binding of receptors and ligands (the reacting sites), and synthetic drug design. Partial volume matching can also aid in the detection of structural motifs (sequences of amino acids that have similar spatial structure) in proteins, thus adding to the understanding of their role and functionality [2]. Yet another motivation is the combination of several snapshots of the same object, taken from different view points, in order to obtain a description of a bigger portion of its boundary. This has obvious industrial, civil, and military applications (e.g., the decoding of aerial photographs in which depth data is assumed to be available), and is closely related to the field of *active vision*, which is currently an intensive topic of research [13]. Another important motivation is the registration of medical images obtained from the same or different modalities. In many cases, more than one imaging technique is used in clinical diagnosis, therapy planning, and in evaluation of therapy. Integrating the complementary information obtained from several studies of the same patient can be a valuable tool in the treatment of the patient. Note that, in most of these applications, we are only seeking a *partial* match between the image and the model objects, or between two protein molecules, or between different views of the same object. Medical image matching, however, usually involves a global match (registration) of a whole organ.

Our work was motivated by earlier works on the *partial curve matching* technique, first proposed by Kalvin et al. [53] and by Schwartz and Sharir [75]. This technique, which uses the so-called *geometric hashing* method, was originally introduced for *curve matching* in the plane. In that problem we are given two curves, such that one is a (slight deformation of a) proper subcurve of the other, and we wish to find the translation and rotation of the subcurve that yields the best least-squares fit to an appropriate portion of the longer curve. The geometric hashing technique was extended and used in computer vision for automatic identification of partially obscured objects in two or three dimensions. See [47], [57],

- *The authors are with the School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. E-mail: {barequet, sharir}@math.tau.ac.il.*
- *G. Barequet is currently a postdoctoral fellow with the Department of Computer Science of Johns Hopkins University.*
- *In addition, M. Sharir is with the Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA.*

Manuscript received 25 July 1995; revised 29 May 1997. Recommended for acceptance by R. Szeliski.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 105207.

0162-8828/97/$10.00 © 1997 IEEE

[61], [62], [63], [84] for various extensions and applications of the technique. Our algorithm also makes intensive use of geometric hashing, but in a somewhat different setup.

1.1 Previous Work

We first briefly review the fairly extensive literature on the problem of surface or volume matching, studied mainly in the context of computer vision and pattern recognition. Some works (e.g., [14]) depend on the ability to match significant features of the objects, like knobs and holes, whose existence is not usually guaranteed. Other methods, which do not rely on the existence of a particular type of features, are *pose-clustering* [78], *alignment* [49], and *geometric hashing*. A comparison between these techniques is found in [83]. Comprehensive surveys on partial surface matching techniques in computer vision are found in [11], [23]. Many other works have addressed the problem, most of which have various limitations. They either restrict the shape of the matched objects (e.g., require them to be polyhedra, or to have large planar portions, or study only planar objects), or assume that there is no occlusion (so a full matching between the objects is sought), or handle only restricted motions, involving fewer than six degrees of freedom. The methods that do not have these restrictions have other disadvantages. For example, some of them are sensitive to statistical outliers, which have to be removed in a preprocessing step. Other methods might converge to a motion that yields only a local extremum of their "scoring function," etc. We provide here, for the convenience of the reader, a quick review of these works.

Potmesil [71], [72] describes a heuristic surface matching algorithm, which searches for the transformation that maximizes shape similarities in the registration of two objects, where the candidate transformations are evaluated at some selected points, e.g., surface control points and points of maximum curvature. Besl [9], [10] gives some metrics for measuring matches between curves and surfaces. Fisher [36] suggests heuristics for obtaining a registration between two objects by using their two-dimensional boundaries (the so-called *silhouettes*). Horn [43] and Brou [18] develop the *extended Gaussian image* method, which uses a surface normal histogram for matching convex (and some restricted nonconvex) shapes. Fang et al. [29] and Stockman and Esteva [79] solve a constrained registration problem between polyhedra, where only two-dimensional translations and rotations are allowed. They extract some edge- and point-features, and accumulate a three-dimensional histogram of possible matches, in which clusters are assumed to indicate possible matches. Faugeras [30] and Faugeras and Hebert [31] use *quaternions* for converting the three-dimensional rotation problem into a four-dimensional minimum eigenvalue problem, while the translation is found by using a standard least-squares technique. Horn [44] suggests instead to look for the maximum eigenvalue. (Although we do not use quaternions, we have used a similar idea for a variant of our approach; see Section 5.) Alternatively, Golub and van Loan [40] and Arun et al. [5] use *singular value decomposition*. The main deficiency of their method is relying on the existence of significantly large planar regions in the objects. Szeliski [80] uses a standard *steepest descent* heuristic for generating a series of rotations of one object relative to the other. His goal is to minimize the sum of weighted differences (along the z-axis only) between points of the two objects. Taubin [81] approximates data point sets with algebraic surfaces up to the 10th degree, with an application to global position estimation (that is, without occlusion). Kamgar-Parsi et al. [54] present a "2.5-dimensional" registration method, which is actually a matching problem in two-space. Besl and McKay [12] register three-dimensional shapes (of various types) by using the so-called *iterative closest point* algorithm. This algorithm iteratively invokes a procedure which finds the closest member of a point set to another given point. The algorithm converges very quickly to a local minimum of a mean-square distance metric, so it is applied from several starting rotations, hoping not to miss the global minimum. Huttenlocher et al. [48] track moving objects in a series of two-dimensional raster images by using the minimum Hausdorff distance under translations between two sets of points. They actually match portions of the two images. Their method assumes that the orientations of occurrences of the same object in successive images differ by only a relatively small amount. This work thus considers only translations, and explicitly assumes that the rotation component of the rigid motion of the object is relatively small. Finally, Lavalee and Szeliski [64] solve the 2D/3D matching problem by performing a least-square minimization of the "energy" needed to bring the projection lines of the camera contours tangent to the object. They do that efficiently by using a precomputed map of *signed distances* which are represented by an octree spline.

First attempts to solve the molecule docking problem, which are based on energy minimization (refs. 1–6 of [55]), were only partially successful. Geometric approaches (refs. 7–16 of [55], including [25]) were much more successful, but (at least the earlier ones) were not reliable enough and suffered from unacceptably long computation time [55]. Kuntz et al. [60] transform the structures of the ligand and of the receptor of two proteins into a graph in which they search for four-cliques. Each detected clique is mapped into a three-dimensional transformation and checked for possible penetration of the ligand into the receptor, in which case it is rejected. Similarly, Kuhl et al. [59] construct a graph in which they search for the maximum clique. Although this problem is NP-complete in general, they claim to obtain a randomized algorithm whose practical running time is $O((nm)^{2.8})$, where n and m are the numbers of atoms in the ligand and the receptor, respectively. Other geometric methods [51], [55] perform a brute-force search over all the discretized three-dimensional rotations, while using a secondary method for identifying the appropriate translation. The paper [55] uses a correlation function (computed efficiently by using the discrete Fourier transform) for determining the translation. Traditional methods for detecting structural motifs in proteins usually employ algorithms for string comparison, where the strings represent the primary structures (amino acid sequences) of the proteins. A survey of these methods is found in [74]. Enhanced methods [1], [67], [73] also consider *predefined* motifs (such as the so-called α-helixes and β-sheets) in the secondary structures of the molecules.

A major contribution to these problems was achieved by application of techniques based on geometric hashing. This method facilitates the handling of a priori totally unknown three-dimensional structures. The main problem in generalizing this technique to partial matching between surfaces (as opposed to curves) is that, in its original application to partial curve matching, the method depends on the linear order of points along the given curves, which is needed for computing the relative "shift" between matching portions of the curves. There are significant technical problems in naive attempts to extend this technique to (partial) matching between surfaces or volumes. In applications based on geometric hashing, one proceeds by assigning *footprints* to the molecule atoms, then by matching the footprints and by voting for the relative transformation (rigid motion) of one molecule relative to the other, assuming that the correct transformation will receive significantly more votes than all the others. For the motif detection, Nussinov and Wolfson [69] define the footprint of each atom as its coordinates in systems defined by any non-colinear triple of atoms (thus each atom has $O(n^3)$ footprints, where n is the number of atoms in the molecule). Similar ideas are presented in [33], [35]. Fischer et al. [34] take a similar approach for the molecule docking problem. In their method, each pair of atoms defines a *basis* (whose length is the distance between the two atoms), and the footprint of every atom is defined as the distances from the atom to the endpoints of every basis, coupled with the length of the basis (thus each atom has $O(n^2)$ footprints). In all cases, the footprints are stored in a hash table, as in any other application of geometric hashing, which allows to retrieve entries with some tolerance. Here this is needed not just because of the noisy footprints, but also because of the conformational changes that might occur in the molecule structures during the reaction between them.

Finally, we briefly describe the topic of medical image matching, which has also attracted a lot of attention in the medical literature. The problem arises when complementary information about some organ is obtained by several imaging techniques, such as CT (Computed Tomography), MRI (Magnetic Resonance Imaging), and others. The goal is to match (register) the various models of the same organ obtained by these methods, in order to obtain a single improved and more accurate model. Such a registration is needed because the orientations of the organ usually differ from one model to another. Many methods, which are similar to the methods for object recognition, were proposed for the solution of this organ registration problem. These include, among many others, approximated least-squares fit between a small number of markers [41], [42], singular value decomposition for matching point pairs [28], [45], high order polynomials for a least-squares fit [56], [76], "thin-plate spline" for registering intrinsic landmarks [16] or extrinsic markers [15], parametric correspondence [22], [70], chamfer maps [7], [17], [26], [46], [50], partial contour matching [68], moments and principal axes matching [3], [37], [38], [58], [82], and correlation functions [6], [20], [24], [52], [66]. Two comprehensive overviews of image registration techniques are given by Brown [19] and by van der Elsen et al. [27].

1.2 Our Approach

We propose a new approach to the matching problem and present several of its applications in the domains mentioned above. Our algorithm accepts any pair of point sets in 3-space, describing either the volumes or the boundary surfaces of two objects, and attempts to find the best rotation and translation of one object relative to the other, so that:

1) if the given sets represent object boundaries, then there should be a good geometric fit between large portions of these boundaries; and
2) if the given sets represent object volumes, then there should be a large fit between the boundaries of the objects, so that their volumes either overlap or remain disjoint near the fit.

In the first case, our algorithm solves the (partial) surface matching problem. In the second case, it solves the (partial) volume matching problem, either with volume overlap or with volume complementarity.

Here is a brief sketch of our algorithm:

- First, we associate with each point of the two sets a *footprint*. This value should be invariant under rotations and translations, and should be "descriptive," in the sense that points of the two sets whose local neighborhoods admit a good match should have similar footprints, whereas points whose local neighborhoods do not fit well together should have significantly differing footprints.
- Next, we define a scoring function that measures the "goodness" of a specific rotation (of one set relative to the other), and is invariant of the relative translation. In an ideal setting, this function has a global maximum at the correct rotation and does not have any other local maxima. This enables us to advance from any rotation toward the correct rotation, by invoking the scoring function iteratively, and by deciding locally in which direction to advance.
- Finally, we compute the best translation associated with the final rotation. The various applications of our algorithm mainly differ in the definition and computation of the footprints. Needless to say, the choice of footprints is a crucial factor that influences the success of our method.

Our technique bears some resemblance to previous indexing methods that are based on the density of votes in some space. The main contribution of our technique is the new observation that the density of votes in translation space can be used for computing the correct relative rotation of a model and an image. Other ingredients of our algorithm are known methods for searching for extremes of unknown functions, clustering, principal components analysis, and several other techniques. In comparing our algorithm to previous works on surface matching, we can say that our algorithm appears to be robust even in the presence of considerable noise in the input data (or of excess data that is irrelevant for the partial match that we seek). Therefore, we do not need to remove, in a preprocessing step, data that represent statistical outliers. Our algorithm does not depend on any correspondence between the two sets of input data points. It does not depend on the existence of any predetermined features of the objects. It does not rely on surface derivatives. Our algorithm is very easy to implement, and it runs in practical time with practical inputs, which compares favorably with reported performance of earlier algorithms. It produced very accurate results in all the cases that we tested and are reported here. However, as other indexing methods, our technique can be fooled by contrived examples (which, however, are unlikely to arise in practice), or when poor footprint systems are being used. The generalization of our algorithm to higher dimensions appears to be straightforward. The only issue preventing the algorithm from being a fully automated tool for matching any two point sets is the need to assign "descriptive" footprints to all the points. This seems to require customized treatment for each class of applications, which may be regarded as a deficiency of our algorithm. However, it is also an advantage of it: Whereas most of the previous registration algorithms actually regard the coordinates of each point as its footprint, we achieve greater versatility of the matching process through the additional information hidden in any specific system of footprints. We emphasize again that the choice of footprints does greatly influence the success of the subsequent matching procedure. As with other indexing methods, our algorithm may fail with a poor choice of footprints. Our experiments show that the algorithm breaks down when the number of correct votes falls below roughly one percent of the total number of votes (as did occur in the molecular matching examples—see Section 7). The largest percentage of correct votes that occurred in our experiments was 16 percent.

In the context of molecular biology, the main differences between the application of our algorithm to molecule docking and similar works on this problem based on geometric hashing are the following:

1) We use all the atoms on the molecule boundaries instead of using only "points (atoms) of interest." (For example, the technique of [34] uses only the "backbone" C-atoms of the polypeptide chain of a protein.)
2) We generate a footprint for each individual atom in each molecule, and not for pairs or triples of atoms. Consequently, we have much fewer footprints than earlier methods do, and thus also much fewer voting entities.
3) The other methods vote directly for a rotation, whereas we vote for an imaginary translation at any fixed rotation. Thus, the other methods have only one voting process, whereas we generate a series of rotations and vote at each one of them separately.

Our technique was most successful in the industrial and medical applications, where the quality of the data allowed us to generate good footprints, and was more problematic (though still reasonably successful) in the molecular biology applications, where the footprints generated by the current version of our algorithm are of poorer quality, due to the nature of the input data. We feel that the difficulties that we faced with molecule docking were not due to inherent limitations of the algorithm, but rather because of the imprecision in the definition and computation of the boundary atoms of the molecules and, consequently, the production of poorer-quality footprints for them. (Needless to say, these problems also hamper the performance of all the earlier algorithms mentioned above.)

The paper is organized as follows. In Section 2 we describe the rationale for the algorithm proposed in the following sections, by examining a two-dimensional variant of the problem. Section 3 presents an overview of the algorithm. Section 4 describes in detail the various phases of the algorithm. Section 5 describes an alternative statistical approach in three dimensions, which reduces the problem, in favorable situations, to a two-dimensional problem. Section 6 analyzes the complexity of the algorithm. Section 7 presents the experimental results mentioned above. We end in Section 8 with some concluding remarks.

2 Rationale: The Two-Dimensional Case

The motivation for our algorithm arose from our experimentation with matching synthetic sets of points in two dimensions. The input consisted of a point set A in $\mathbb{R}^2$, and of another point set B, obtained from A by rotating (by some angle θ) and translating. The footprints of the points were chosen in an artificial way that ensured a nearly perfect match. Consider the point set A shown in Fig. 1. As seen in the figure, the points belong to a regular grid. Each point was assigned a unique footprint. Then, we repeated the following step for various angles θ. The set A was rotated counter-clockwise about the origin by θ. Denote this rotation as R_θ. The new set B was defined as the collection of all the rotated points, where the coordinates were rounded to the nearest grid point. (When more than one such rotated point was rounded to the same grid point, only one representative point was arbitrarily selected.) Each point $R_\theta(p)$ was assigned the same footprint as p. Now we intentionally assumed the wrong assumption that B was obtained from A by *translating* by some shift (t_x, t_y) instead of by rotating. Under this assumption we voted for the relative shift between A and B in the following manner. For each point $p \in A$ we located the point $q \in B$ with the same footprint (if it existed) and voted for the shift $\vec{q} - \vec{p}$. Had our assumption been true, all the votes would have been given to the same shift (t_x, t_y) (or, because of our coordinate rounding, to shifts very close to (t_x, t_y)). Since it was false, the votes were not given to one single cell (shift) but were spread in the voting table. Fig. 2 shows several voting tables that correspond to different values of θ. The voting tables show that the scattering of the votes increases as the angle of rotation increases, reaching a maximum at $\theta = \pi$. Surprisingly, the distribution of the votes in the voting table resembles a rotated version of the original set A! This is easily confirmed when we calculate the shift for which each point $p = (x, y)$ voted. Let S_f denote the scaling of both axes by f. Then, it is easy to see that, for any point $p(x, y)$,

$$
\begin{aligned}
\vec{q}-\vec{p} &= \overrightarrow{R_\theta(p)}-\vec{p}\\
&= (x\cos\theta - y\sin\theta,\ x\sin\theta + y\cos\theta) - (x, y)\\
&= \big(x(\cos\theta - 1) - y\sin\theta,\ x\sin\theta + y(\cos\theta - 1)\big)\\
&= \begin{pmatrix} -2x\sin^2\frac{\theta}{2} - 2y\sin\frac{\theta}{2}\cos\frac{\theta}{2}, \\ 2x\sin\frac{\theta}{2}\cos\frac{\theta}{2} - 2y\sin^2\frac{\theta}{2} \end{pmatrix}\\
&= 2\sin\frac{\theta}{2}\cdot\left(-x\sin\frac{\theta}{2} - y\cos\frac{\theta}{2},\ x\cos\frac{\theta}{2} - y\sin\frac{\theta}{2}\right)\\
&= 2\sin\frac{\theta}{2}\cdot\begin{pmatrix} x\cos\left(\frac{\pi}{2}+\frac{\theta}{2}\right) - y\sin\left(\frac{\pi}{2}+\frac{\theta}{2}\right), \\ x\sin\left(\frac{\pi}{2}+\frac{\theta}{2}\right) + y\cos\left(\frac{\pi}{2}+\frac{\theta}{2}\right) \end{pmatrix}\\
&= S_{2\sin\frac{\theta}{2}}\left(R_{\frac{\pi}{2}+\frac{\theta}{2}}(p)\right)
\end{aligned}
$$

Hence, if the rotation of B relative to A is θ, then each point p votes for the imaginary shift $\vec{q}-\vec{p}$, which is obtained by first rotating p around the origin by $\frac{\pi}{2}+\frac{\theta}{2}$, and then by scaling it by $2\sin\frac{\theta}{2}$. This means that the voting table actually shows the shape of the original set rotated by $\frac{\pi}{2}+\frac{\theta}{2}$ and scaled by a factor of $2\sin\frac{\theta}{2}$.

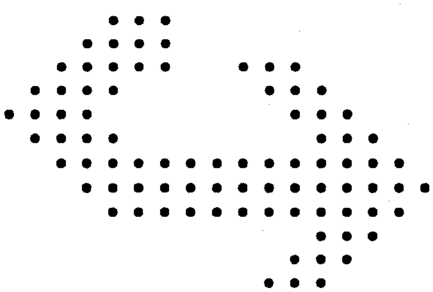


Fig. 1. Synthetic two-dimensional point set.

This suggests the following technique for determining the goodness of a rotation ω between A and B. We rotate A by ω and vote for the shift between $R_\omega(A)$ and B. The closer ω is to the correct rotation θ, the more "compact" is the resulting voting table. That is, the votes should appear to be clustered around some "accumulation point." As shown above, when the relative rotation between A and B is θ, each point $p \in A$ contributes one vote to the shift $S_{2\sin\frac{\theta}{2}}\left(R_{\frac{\pi}{2}+\frac{\theta}{2}}(p)\right)$. Apart from round-off errors, the rotation $R_{\frac{\pi}{2}+\frac{\theta}{2}}$ does not influence the density of the voting table. Thus, the only factor that causes the votes not to be gathered at a single cell of the voting table is the scaling effected by $S_{2\sin\frac{\theta}{2}}$. Since the sine function $\sin\frac{\theta}{2}$ is continuous over the space of orientations $\theta \in [0, 2\pi]$ and has a unique minimum (at 0) and a unique maximum (at π), we expect the voting table to be the most sparse when our "guess" ω deviates from the real θ by π, i.e., when $|\theta - \omega| = \pi$, to be the most dense when $\theta = \omega$, and to vary continuously and monotonically between these two extremes. This suggests that we use a scoring function that measures the *sparseness* of the voting table, giving higher scores to more compact tables, and then apply a simple iterative binary search step that varies the rotation in the direction that makes the table more compact. A plausible scoring function of this kind is

$$SC_0(T) = \sum_{i=1}^{n} M_i^2$$

where T is the voting table, n is the number of cells in T, and M_i is the number of votes given to the ith cell. The performance of this scoring function was very good in our experiments in two dimensions, as well as in our experiments in three dimensions which involved full matching of data free from noise. It failed, however, in cases of partial matching, or when the data was noisy. We describe in detail in Section 4.2 the improved scoring function that we actually used in our experimentation.

3 Overview of the Algorithm

After presenting the basic idea of the algorithm in an ideal and artificial two-dimensional setting, we now extend this idea and develop from it the actual algorithm that we have used. In a nutshell, the idea is to separate, as above, the rotation and translation components of the desired rigid transformation, to conduct a search only over the space of rotations, and to compute a score for each rotation, based on an attempt to compute the correct translation under the (usually false) assumption that the current rotation is the correct one. We are given two sets of points (not necessarily of equal sizes) representing two respective objects in three-space, and expected to be spread more or less uniformly on the boundary of the corresponding objects or in the volumes that they occupy. In the former case we seek a partial (or full) surface match between the boundaries of the two objects, whereas in the latter case we seek a volume match, involving either volume overlap or volume complementarity. Our proposed algorithm consists of the following steps:

1) Data acquisition:
   - Read all the input points describing the two objects. Optionally (in difficult cases), discard points which do not contribute to the matching (e.g., because their footprints are "insignificant").
   - Compute a footprint for each input point. Points that are expected to match (locally) should have similar footprints, and points that should not be matched (locally) should have significantly different footprints.
   - Prepare a *generic voting list*. That is, construct a list of pairs of points, one of each object, such that the

	15
-25	88

(a) $\theta = 0$

	13	14	15
-22		4	
-23	8	25	6
-24	11	18	
-25	6	10	

(b) $\theta = \frac{\pi}{18}$

	08	09	10	11	12	13	14
-18			3	1	1		
-19			4	4	4	1	
-20		3	2	4	1	3	2
-21		4	1	4	2		1
-22		1		3	3		
-23		2		2	4		
-24	2	4	1	3	4		
-25	1	1	4	4	1		
-26			1	2			

(c) $\theta = \frac{\pi}{6}$

	01	02	03	04	05	06	07	08	09	10	11
-14			1	1	1						
-15		1	1	1	1	1	1	1		1	
-16		1	1	1	1	1	1	1	1	1	
-17		1	1		1	2			1		1
-18		1	1	2	1		1				
-19	1	1	1		1	1	1				
-20	1	1			1	1		1			
-21	1	1				1	2	1			
-22		1				1		1	1		
-23							1	1	1		
-24							1		1	1	
-25		1	1				1	2	1		
-26		1	1	1	1		2		1		
-27		1	1	1	1	1	1	1			
-28			1	1	1	1		2			
-29							1	1			

(d) $\theta = \frac{\pi}{3}$

	-6	-5	-4	-3	-2	-1	00	01	02	03	04	05	06	07
-14	1		1		1		1		1		1			
-15		1		1		1		1		1		1		
-16	1		1		1		1		1		1		1	
-17		1		1		1								
-18	1		1		1		1							
-19		1		1		1		1						
-20	1		1		1		1		1					
-21		1				1		1		1				
-22	1		1				1		1		1			
-23		1						1		1		1		
-24	1		1						1		1		1	
-25		1								1		1		1
-26			1								1		1	
-27												1		1
-28											1		1	
-29						1						1		1
-30					1		1		1		1		1	
-31				1		1		1		1		1		1
-32					1		1		1		1		1	
-33						1		1		1		1		1

(e) $\theta = \frac{\pi}{2}$

	..	..	..	..	..	..	..	..	-9	-8	-7	-6	-5	-4	-3	-2	-1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
-25											1		1		1																		
-26																																	
-27									1		1		1																				
-28																																	
-29							1		1		1																						
-30																																	
-31					1		1		1																								
-32																																	
-33			1		1		1		1		1		1		1		1		1		1		1		1								
-34																																	
-35	1		1		1		1		1		1		1		1		1		1		1		1		1		1						
-36																																	
-37			1		1		1		1		1		1		1		1		1		1		1		1		1		1				
-38																																	
-39					1		1		1																1		1		1		1		
-40																																	
-41							1		1		1																1		1		1		1
-42																																	
-43									1		1		1												1		1		1		1		
-44																																	
-45											1		1		1						1		1		1		1		1				
-46																																	
-47																					1		1		1		1						
-48																																	
-49																					1		1		1								

(f) $\theta = \pi$

	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
-33												1		1		1		1		1
-34			1								1		1		1		1		1	
-35		1								1		1		1		1		1		1
-36	1		1						1		1		1		1		1		1	
-37		1						1		1		1						1		1
-38	1		1				1		1		1						1		1	
-39		1				1		1		1								1		1
-40	1		1		1		1		1								1		1	
-41		1		1		1		1								1		1		1
-42	1		1		1		1										1		1	
-43		1		1		1												1		
-44	1		1		1		1		1		1		1							
-45		1		1		1		1		1		1								
-46	1		1		1		1		1		1									

(g) $\theta = \frac{3\pi}{2}$

	14	15	16	17	18	19	20
-26					1	1	
-27				1	4	4	1
-28			2	4	1	4	3
-29			2	4		2	4
-30	3		4	2			1
-31	4	2	4	2	1	2	
-32	1	4	4	3	3	1	
-33		3	3	2			
-34		1					

(h) $\theta = \frac{11\pi}{6}$

Fig. 2. Voting tables corresponding to a few values of θ.

difference between the footprints of the points in a pair does not exceed some tolerance threshold.

2) Scoring a specified rotation R:

- Vote for the translation between the objects. For each pair of points (p, q) in the generic voting list, apply the rotation R to p and give one vote for $\vec{q} - R(p)$.
- Compute a score for the resulting voting table. This function aims to give higher scores to more-clustered voting tables.

3) Advancing toward the rotation with the highest score:

- Compute the scores of rotations defined by a sparse three-dimensional grid of Euler angles taken at fixed intervals. Initialize the current rotation to the rotation on this grid that receives the highest score.
- Advancing in large steps: Initialize d to d_0. Compute the scores of all the rotations (in some predefined grid) that are at some Manhattan distance d from the current rotation, and find their maximum. If the maximum score is higher than the score of the current rotation, reset the current rotation to

the one with improved score, reset d to d_0, and repeat this step. Otherwise, double d and repeat this step. Halt this step when d exceeds some limit $d_{\max}$.

- Advancing in small steps: Initialize d to $d_0/2$. Compute the scores of all the rotations in a grid of width $d/2$ that are at Manhattan distance d from the current rotation, and find their maximum. If it is higher than the score of the current rotation, reset the current rotation to the one with improved score, double d (unless it equals $d_0/2$), and repeat this step. Otherwise, divide d by two and repeat this step. Halt this step when d falls below some limit $d_{\min}$.

4) Computing the correct translation:

- Find the cell with the maximum number of votes in the voting table associated with the best rotation, and declare it to be the correct translation. In difficult cases, where too much noise causes this simple approach to fail, invoke a special subroutine that uses a correlation function to compute the correct translation.

4 Detailed Description of the Algorithm

4.1 Data Acquisition

The actual data consists of two point sets $A, B \subset \mathbb{R}^3$. The first step computes the footprint of each point, as a certain scalar or vector function of the points of the same set lying in some small neighborhood of the point. In practice, different types of input data require different, and in many cases rather careful, definition and computation of the footprints. The footprint should be invariant under translations and rotations in three dimensions, and also be sufficiently "descriptive," as described above. Intuitively, a good choice of footprints leads to a small number of false matches, i.e., pairs of points, one of each set, which have similar footprints although they should not match. Our experimentation showed that avoiding false matches is much more important (and difficult) than not losing true matches (the latter problem can result from sporadic mismatches between footprints of points that are supposed to match). We denote the footprint of every point p by FP(p). For each point $p \in A$, we find all the points $q \in B$ whose footprints are close enough to that of p, that is, $|\,\mathrm{FP}(p) - \mathrm{FP}(q)\,| \le \varepsilon$ (for some tolerance parameter ε). Each such pair (p, q) contributes one vote for the relative translation $\vec{q} - \overrightarrow{R(p)}$ between the sets A and B. Clearly, the set $\{(p, q)\}$ of voting pairs is independent of the rotation R. Therefore, we preprocess this set in advance. First, we prepare the set FP(B) for *range searching* in the footprint space. Then, for each point $p \in A$, we generate a range-searching query consisting of the ball of radius ε about FP(p), and, for each point $q \in B$ found in this range, we add the pair (p, q) to the set of voting pairs, denoted as the *generic voting list*.

4.2 Scoring a Rotation

We represent a rotation R of the set A by a triple of *Euler angles* $(\theta_x, \theta_y, \theta_z)$ (see [39, p. 608]). That is, every point $p \in A$ is first rotated by θ_x around the x-axis, then rotated by θ_y around the y-axis, and finally rotated by θ_z around the z-axis. (Equivalent definitions are found elsewhere, e.g., in [11, p. 79].) For each rotation R, we scan sequentially the generic voting list, compute the actual value of $\vec{q} - \overrightarrow{R(p)}$ for each voting pair (p, q), round its three components to integer numbers, and give the resulting triple one vote. The resulting voting table T_R is thus three-dimensional and contains some number, n_R, of nonempty cells. Denote the number of votes in the ith cell by M_i, for $i = 1, \ldots, n_R$. Also, denote the Euclidean distance between two cells i and j by D_{ij}, and set the "radius" of every cell to $R_i = 1.0$. The first version of our scoring function is similar to a measure of a "gravity potential." It regards each cell of the voting table as a volume, whose mass is the number of votes given to it. In order to favor dense voting tables, the score consists of the sum of the "gravity potentials" between every pair of distinct cells, plus the "self gravity potential" of each cell. Specifically, this score is defined as

$$SC_1(T_R) = \sum_{1\le i<j\le n_R} \left(M_i M_j / D_{ij}\right) + \sum_{i=1}^{n_R} \left(M_i^2 / R_i\right)$$

The score SC$_1$ performed better than SC$_0$ (see Section 2) in two and in three dimensions (in the sense that the correct rotation corresponded to a sharper maximum of the scoring function in the noiseless cases that we have tested), but still did not score well the correct solutions in typical cases that involve noise. We found empirically that we can significantly improve the score if we weigh each term of SC$_1$ proportionally to the masses that participate in it. That is, we used the scoring function

$$SC_2(T_R) = \sum_{1\le i<j\le n_R} \left(M_i M_j \left(M_i + M_j\right) / D_{ij}\right) + \sum_{i=1}^{n_R} \left(M_i^3 / R_i\right)$$

For efficiency of computation, we restricted ourselves to sufficiently "close" cells of the voting table, i.e., to cells whose mutual Manhattan distance did not exceed some threshold δ. In all our experiments we set $\delta = 5$. The effect of this compromise on SC$_2$ was negligible.

In abstract setting, the problem that we face here is as follows. We are given a three-dimensional distribution (the voting table), and we want to detect in it a cluster region, where the distribution is denser and more concentrated. We want to give a higher score to distributions with a denser cluster. As explained in Section 4.3 below, the dense cluster, when it exists, fully lies in (or, because of noise and of rounding, near) some plane, a fact that should help us in its detection. However, we are not aware of any available statistical tools that can tackle this problem successfully on the type of data that we have.

4.3 Finding the Best Transformation

Recall that, in two dimensions, the votes have the simple representation

$$\overrightarrow{R_\theta(p)} - \bar{p} = S_{2\sin\frac{\theta}{2}}\left(R_{\frac{\pi}{2}+\frac{\theta}{2}}(p)\right)$$

Fortunately, a similar situation also occurs in three dimensions. As is well known, every rigid motion in three dimensions can be represented by a single rotation by some angle θ around some line ℓ in $\mathbb{R}^3$ (which passes through the origin), followed by some translation s. Without loss of generality, we may define a new coordinate system, in which the z'-axis is the line ℓ. In this system, voting for $\overrightarrow{R_\theta(p)} - \bar{p}$ is identical to the two-dimensional case. Indeed, rotating a point by angle θ around the z-axis does not change its z-coordinate, so the z-component of the vote is simply zero. Hence, the "good" votes in the resulting three-dimensional voting table fully lie within a plane Π, which is perpendicular to the direction ℓ of rotation and passes through s. The structure of the "good" votes within this plane is essentially identical to the structure of an ideal two-dimensional voting table of the orthogonal projections of the given points on Π, which corresponds to the planar rotation by angle θ. For a given rotation R, let $\Theta(R)$ denote the angle of rotation effected by R about the rotation axis of R. If we assume perfect point registration and no noise in our match (as we assumed in the two-dimensional case), then the above discussion implies that, with an ideal choice of the scoring function, we would obtain a maximum score when the current rotation R is equal to the correct rotation R_0, and the score would increase monotonically toward that maximum as the angle $\Theta(R_0R^{-1})$ decreases to zero. In practice, however, due to imperfect registration of the footprints, the existence of only a partial match, and noise in the data, even an ideal scoring function might have other local maxima, but we expect that these maxima are not too sharp, and that the function resumes its ascent toward the correct rotation in some small neighborhood of any local maximum. These expectations were indeed fulfilled in all our experimentations, when a reasonable (partial) match did exist between the two objects.

The navigation toward the optimum is a combination of a *steepest descent* [77] and a *hierarchical pyramid* [17] approaches. It is performed in three main phases. First, we evaluate the scores of all the rotations $(\theta_x, \theta_y, \theta_z)$, where θ_x, θ_y, and θ_z range over all possible multiples of $\pi/3$. It is well known that $(\theta_x, \theta_y, \theta_z) \equiv (\pi + \theta_x, \pi - \theta_y, \pi + \theta_z)$. Hence we need to score only 108 different rotations (instead of 216) in this phase. We choose the rotation that receives the highest score in this phase as the starting rotation. In cases when multiple solutions (maxima) are expected, or when more than one rotation receives in this step a high score, we repeat the following two steps for several starting rotations. In the second phase we advance in "large" steps. We set two integer values, d_0 and $d_{\max}$ (or require them from the user), which limit the Manhattan distance (in the three-dimensional space of Euler's angles) from the current rotation in our search for an improved rotation. Assume that the current rotation is $(\theta_x, \theta_y, \theta_z)$. We initialize d to d_0 and compute the score of all the rotations $(\theta_x + \varepsilon_x, \theta_y + \varepsilon_y, \theta_z + \varepsilon_z)$, where ε_x, ε_y, and ε_z are integers, and $|\varepsilon_x| + |\varepsilon_y| + |\varepsilon_z| = d$. If the highest score of all these rotations is larger than that of the current rotation, we reset the current rotation to the new one, reset d to d_0, and repeat this step. Otherwise, we double the value of d and repeat this step. If the value of d exceeds $d_{\max}$, then we halt this phase and proceed to the third phase. In the third phase we advance in "small" steps. We set a value $d_{\min}$ which, as in the previous phase, limits the Manhattan distance from the current rotation in our search for improved rotations. The initial rotation is set to the last one of the previous phase. We initialize d to $d_0/2$, and compute the score of all the rotations $(\theta_x + \varepsilon_x, \theta_y + \varepsilon_y, \theta_z + \varepsilon_z)$, where ε_x, ε_y, and ε_z are multiples of $d/2$, and $|\varepsilon_x| + |\varepsilon_y| + |\varepsilon_z| = d$. If the highest score of all these rotations is larger than that of the current rotation, then we reset the current rotation to the improved one, double d (unless it equals $d_0/2$), and repeat this step. Otherwise, we divide d by two and repeat this step. If the value of d falls below $d_{\min}$, then we terminate this part of the algorithm and output the current rotation.

4.4 Determining the Correct Translation

In most cases it is very simple to infer the correct translation from the voting table of the best rotation. It is simply the cell that receives the highest number of votes. It can be regarded as the "center" of the cluster representing the correct translation. This choice of translation was confirmed by most of our experiments. However, this assumption turned out to be false for voting tables that contained too much noise (due to false votes, noisy data, and/or only partial matches) and were too sparse (due to inaccurate choice of footprints and/or a tolerance parameter). Such a situation arose in our experimentation with molecule docking (see Section 7). It is interesting that in these bad cases the voting tables were descriptive enough for yielding the correct rotation, but too noisy to point directly at the correct translation. Our explanation of this phenomenon is that in each voting table the correct votes were too few to be recognized; however, when we advanced from one rotation to a more accurate rotation, the few correct votes approached each other so as to give a higher score to the table that corresponded to the latter rotation. In such problematic cases we use a correlation function, such as the function described in [55]. In this simple approach we rely on the ability to distinguish between the boundary of the object and the internal portion of it. We define for the first point set A a characteristic function of the xyz-grid points: $f_A(x, y, z)$ is set to one for points on the boundary of the object, to c_A for points inside the object, and to zero for points outside the object. We define a similar characteristic function f_B for the point set B, using another constant c_B. The constant c_A (resp. c_B) is chosen to be a large negative (resp. small positive) number. Then, we evaluate for each translation $t = (t_x, t_y, t_z)$ (in a predefined range of translations) the correlation function defined by $\mathrm{COR}_{AB}(t_x, t_y, t_z) = \sum_x \sum_y \sum_z (f_A(x, y, z) \cdot$

$f_B(x + t_x, y + t_y, z + t_z))$, and search for the translation t^* which maximizes this correlation. This t^* is output as the correct translation.

5 An Alternative Statistical Approach

In this section, we describe a statistical approach, which is capable of reducing the three-dimensional matching problem into a two-dimensional problem. We note in advance that, so far in our experimentation, this method succeeded only in a favorable situation, where we had a very good footprint system, in which the amount of false votes in the voting tables was sufficiently small. However, when this technique applies, it improves the running time of the algorithm considerably, because it constrains significantly the search for the best rotation.

Let X denote a three-dimensional random variable, and let Σ denote the covariance matrix of X; that is,

$$\Sigma_{i,j} = \left(\sum_{k=1}^{N}\left(X_{ik} - \overline{X_i}\right)\left(X_{jk} - \overline{X_j}\right)\right)\Big/(N-1)$$

for $i, j = 1, 2, 3$, where N is the number of samples of X, X_{ik} is the ith component of the kth sample of X, and $\overline{X_i}$ is the average of the random variable X_i obtained by projecting X onto the ith axis. Let v_1, v_2, and v_3 be the characteristic vectors of Σ, with corresponding characteristic values $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. The vectors v_i are known as the *principal components* of X. They have the property that the projection of X onto the line containing v_1 gives a one-dimensional variable with the largest variance. The direction v_2 yields the maximum variance for the projection of X in any direction orthogonal to v_1, and v_3 yields the maximum variance for the projection of X in any direction orthogonal to both v_1 and v_2. See [4, §11] for more details concerning principal components.

Consider the result of scoring some fixed rotation, say (0, 0, 0). As explained in Section 4.3, all the "good" votes are spread in a plane Π, which is perpendicular to the axis of rotation ℓ (about which the second point set was rotated relative the first set). If there are significantly many "good" votes, we can perform a principal components analysis on the resulting voting table and find the plane Π. In a perfect situation, when all the votes are indeed spread in a plane Π, the variance of the voting table in the direction of the line ℓ orthogonal to Π is zero. Hence, in this case the two most significant principal components span the plane Π, and the least significant component (with characteristic value zero) gives the direction of ℓ. In more realistic noisy situations, where the amount of noise present in the voting table is not too large, this method still gives a good approximation to the direction of ℓ. Next, given ℓ, we generate a transformation matrix T for which the image of ℓ is the z-axis. If we apply T to the original data and use these transformed values in subsequent voting steps, we obtain voting tables where (most of) the votes lie in some horizontal plane. We can thus reduce the problem to a two-dimensional problem by projecting the data onto the xy-plane. We can now compute the desired rotation angle θ (about ℓ) by the simple method described in Section 2. The rotation in our original three-dimensional problem is thus $T^{-1}R_\theta T$, where R_θ is the two-dimensional rotation about the origin by θ. We thus obtain the rotation component of the solution. (Note that the rotation is represented by a rotation matrix and not by Euler angles.) We can now find the translation component of the solution by applying a single voting step (of our method) to the correct rotation, and by looking for the cell that received the largest amount of votes.

This technique resembles the approaches taken in [3], [31], [44], [58], [82]. The major difference in our approach is that we use this technique only for reducing the dimensionality of the problem by applying it to our voting table, while the works cited above usually attempted to compute directly the correct rotation by aligning the principal axes of inertia (equality of the second order moments) of the original data. This latter approach is more ambitious, and can succeed, as noted in the introduction (Section 1), only when the amount of statistical outliers is negligible. The major disadvantage of this approach is that it requires a very good footprint system, or, alternatively, a preprocessing step that removes noise from the input data or from the initial voting table. Such a good footprint system, in which the number of "bad" votes is negligible, is not always practical (or possible) to design. Moreover, we do not have a good method for removing noise from the voting table; this lies at the heart of the whole problem—the availability of such a cleaning step would have made surface matching a much simpler task. Thus, in favorable situations, in which the "good" votes dominate the "bad" votes, the principal components technique works well and yields a much faster algorithm, in which ℓ is found in a single step, followed by a one-dimensional search for the rotation angle about ℓ, thereby eliminating the more expensive search in the three-dimensional space of rotations. In practice, however, the voting tables are so noisy, that this computation of ℓ cannot be performed successfully on any single voting table. The next section describes one successful application of this technique.

Note that this mechanism works only when the relative rotation between the two point sets is "far" from the rotation for which the voting table is constructed. Otherwise the variances of the two major principal components decrease, and they may be indistinguishable from the small variance along ℓ. In this case all the "good" votes are accumulated around some point, which is the correct shift. Since we do not know in advance how close the correct rotation to an initial rotation is, we can simply apply this technique to several rotated versions of the first point set, which are relatively far from each other. Most of these rotations will be far from the correct rotation, so we can eliminate the rotations with the smallest overall variance and average the results of all the others, thereby obtaining even more robust determination of ℓ. It is also possible to combine the statistical method with the iterative method. This is performed by applying the third phase of our standard advancing mechanism (see Section 4.3) after applying the mechanism presented here. This way we combine a fast computation of an approximate solution (by using the statistical approach), with a more precise mechanism for reaching closer to the correct solution (by using our regular advancing method). Another possible combination is to replace the first phase of our standard method (the scoring of a grid of rotations) by the statistical method. However, in the one successful application of the statistical technique, the result was accurate enough and did not require further improvement. The statistical approach saved in this case more than 90 percent of the running time.

6 Complexity Analysis

We measure the time complexity of the algorithm as a function of n, the cardinality of the input sets (assuming the two input sets have comparable sizes), k, the size of the generic voting list, and s, the maximum size of the object in any direction, which is proportional to, and is measured in terms of the maximum size of any dimension of the voting table. Computing the footprints takes $O(n)$ time. Preparing the generic voting list, if it uses a hash table, can be executed with expected $O(k)$ running time. This expected running time is due to the nature of hashing, and does not assume anything about the input point sets of the algorithm. A reasonable choice of the footprint system and of the tolerance parameter ε, which yields on average only a constant number of points of the second set for each range-searching query made for a point of the first set, makes k comparable with n. Improper choice of the footprints and/or of ε, say a too large ε, might result in $\Theta(n^2)$ access operations to the hash table, which might make k also $\Theta(n^2)$ in the worst case. (We could also achieve an $O(n\log^2 n + k)$ deterministic running time by using fractional cascading [21].) Each scoring step consists of two substeps:

- Creating the actual voting table, which takes $O(k)$ time; and
- Scoring the voting table, which takes $O(s^3)$ time (since for each cell of the table we consider only a constant number of neighboring cells).

The number of scoring operations is proportional to $1/d_{\min}$. (With our setting of the tuning parameters, the maximum value of this term was a few thousands.) In total, the running time of the whole algorithm, with an appropriate tuning of its parameters, is $O(n + k + (k + s^3)/d_{\min})$. The space complexity of the algorithm is $O\,(n + k + s^3)$.

7 Experimental Results

We have implemented the whole algorithm in C on a Digital DECstation 5000/240, on a Sun SparcStation II, and on SGI Indigo and Indy workstations. The implementation, performed by the first author, took about two man months, and the software consists of about 2,000 lines of code.[1] We have experimented with the algorithm on several data files obtained from different sources of input, each input consisting of 300–1,200 points, and obtained very good results in practically all cases. We developed specific methods for computing "meaningful" footprints for each type of input, as described below. Also, for each type of input we found (empirically) a tolerance parameter for preparing the generic voting list. The tuning of the parameters that controlled the advancing phases (limits of rotation distances) was very robust, and a single set of values performed well in all cases. We used $d_0 = 4.0$ (resp. $d_{\max} = 16.0$) as the lower (resp. upper) limit for advancing in large steps, and $d_0/2 = 2.0$ (resp. $d_{\min} = 0.125$) as the upper (resp. lower) limit for advancing in small steps. All the angles are specified in degrees.

In the first example, we sought a full volume match between two versions of a CAD object. Fig. 3a shows the so-called *Geneva* mechanism [8, pp. 679–680]. This object was rotated by the Euler angles (15.7, 115.2, 200.1), as shown in Fig. 3b. Then, the two objects were approximated by rather large voxels, as shown in Fig. 3c and Fig. 3d, respectively. The size of each voxel was set to one, and the point sets representing the two objects were defined to consist of the center of each voxel. The footprints chosen for this example count the "amount of material" around each voxel: It is the number of voxels belonging to the object within the 5 × 5 × 5 cube centered at the given voxel. The tolerance parameter for the voting mechanism (maximum tolerated difference between footprints) was set to one. The algorithm found the rotation within error of less than 0.8 degrees at each component. The algorithm also found the correct translation. Fig. 3e shows an overlay of the two objects, where the first one is rotated and translated according to the computed transformation. See also Table 1 for more performance details.

In the next example, we looked for a partial surface match between two high-resolution depth-sensing scans of a car from two view points, obtained by a commercial digitizer. Figs. 4a and 4b show two "clouds" of points, which contain 119,290 and 179,216 points, respectively. Because of the nature of the scanning, the two sets of points describe xy-monotone surfaces (each relative to a different coordinate frame). Fig. 4a (resp. 4b) shows a scan of the left side (resp. front half) of the car, as seen from behind (resp. the front of) the car. We chose one representative point out of every 10 × 10 square of points in the respective xy-grids: the average of the coordinates of the points. Thus, the size of the data that we considered was only $\frac{1}{100}$ of the original data. Figs. 4c and 4d show the two new images from the same view point. The footprints chosen for this example aim to encode the "pyramid" of material emanating from each vertex $p_{i,j}$ of the new grid of data points, and is given by the average of four spatial angles around $p_{i,j}$:

[1] The software is publicly available and can be obtained upon request from the first author.

TABLE 1
PERFORMANCE OF THE MATCHING ALGORITHM

Case	Figure	Matching Type	|*A*|	|*B*|	Footprint	Tolerance Parameter	Voting Pairs
Geneva	3	Volume	553	552	# of voxels in a 5-box	1	16,721
Car	4	Surface	1,185[1]	1,728[1]	Surface curvature	1	13,609
Aircraft	6	Surface	1,336[2]	1,157[2]	Spatial angles	0	6,438
CAD model	8	Surface	440	(same)	Spatial angles	0	6,636
Molecules (2HHB)	9	Volume	1,112[3]	1,166[3]	# of occupied Å^3-cells in a $9 \times 9 \times 9$ box	4	6,242
Phantom	11	Surface	277	272	# of occupied cells in a $5 \times 5 \times 5$ box	1	4,236
		Surface	277	295		1	4,432
		Surface	272	295		1	4,476

Case	Correct Rotation	Correct Translation
Geneva	(15.7, 115.2, 200.1)	(3, 11, 20)
Car	Unknown	Unknown
Aircraft	(0.0, 0.0, 0.0)	Unknown
CAD model	(180.0, 0.0, 0.0)	(0, 0, 0)
Molecules (2HHB)	Unknown	Unknown
Phantom (1 & 2)	(0.0, 0.0, 0.0)	(0, 0, 0)
Phantom (1 & 3)	(–8.0, 0.0, 0.0)	~(0, 0, 0)
Phantom (2 & 3)	(–8.0, 0.0, 0.0)	~(0, 0, 0)

Case	Computed Rotation		Computed Translation(s)
	Phase II	Phase III	
Geneva	(16.0, 116.0, 200.0)	(16.0, 115.625, 200.875)	(3, 11, 20)
Car[4]	(6.0, 4.0, 182.0)	(6.000, 4.875, 180.875)	(16, 41, 0)
Aircraft[4, 5]	(0.0, 0.0, 0.0)	(–1.5, –1.0, 0.25)	(0, 32, 0), (0, 40, 0), (45, 36, 8), (–45, 36, –8)
CAD model[4, 5]	(180.0, 0.0, 0.0)	(180.0, 0.0, 0.0)	(0, 0, 0)
Molecules (2HHB)	(0.0, 180.0, 180.0)	(–0.125, 181.0, 181.375)	(1, –2, 2)
Phantom (1 & 2)[5]	(0.0, 0.0, 0.0)	(0.0, 0.0, 0.0)	(0, 0, 0)
Phantom (1 & 3)[5]	(–8.0, 0.0, 0.0)	(–8.25, –0.25, 0.0)	(0, –1, 2)
Phantom (2 & 3)[5]	(–8.0, 0.0, 0.0)	(–8.00, 1.5, 1.0)	(0, –1, 2)

Case	# of Scoring Operations				Time (Sec.)				
	I	II	III	Total	I	II	III	Total	Per S.O.
Geneva	108	288	270	666	57	146	137	340	0.51
Car	108	108	162	378	56	53	79	188	0.50
Aircraft	108	36	198	342	89	24	132	245	0.72
CAD model	108	36	72	216	37	12	24	73	0.34
Mol. (2HHB)	108	72	288	468	87	58	233	378	0.81
Pha. (1 & 2)	108	36	54	198	61	20	27	108	0.55
Pha. (1 & 3)	108	72	126	306	60	36	73	169	0.55
Pha. (2 & 3)	108	72	108	288	56	41	64	161	0.55

[1] *433 and 605, respectively, after omitting "flat" points.*

[2] *514 and 497, respectively, after omitting "flat" points.*

[3] *473 and 480, respectively, after eliminating "inner" points.*

[4] *In this experiment* d_{min} *was set to 0.25.*

[5] *In these experiments* d_{max} *was set to 8.0.*

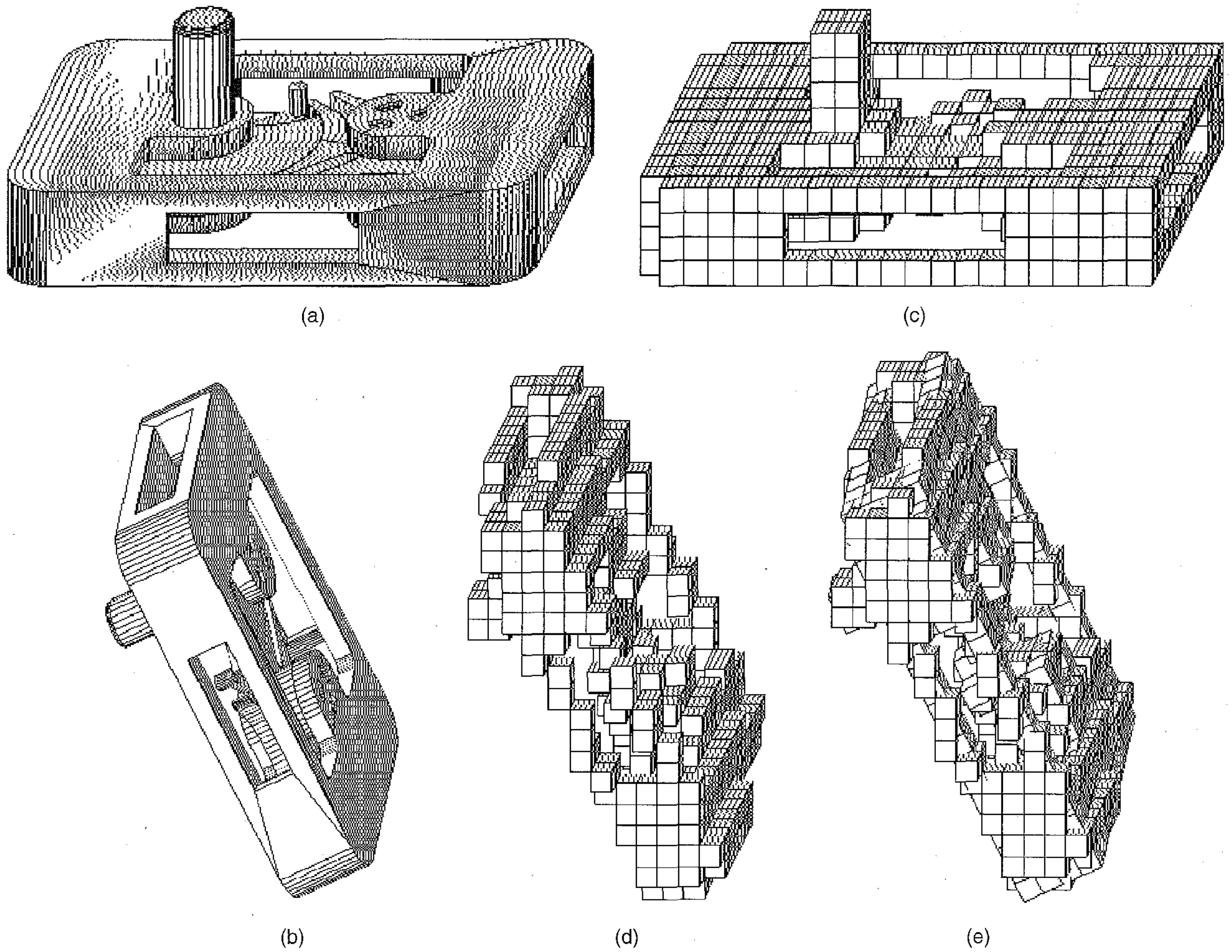


Fig. 3. A full volume matching of CAD data.

$$\mathrm{FP}\left(p_{i,j}\right)=\frac{1}{4}\begin{pmatrix}\angle p_{i-1,j}p_{i,j}p_{i+1,j}+\\ \angle p_{i-1,j+1}p_{i,j}p_{i+1,j-1}+\\ \angle p_{i,j+1}p_{i,j}p_{i,j-1}+\\ \angle p_{i+1,j+1}p_{i,j}p_{i-1,j-1}\end{pmatrix}$$

This footprint approximates the surface curvature at each point, which fits our goal here: surface rather than volume matching. It distinguishes well between peaks in the surface and flat areas, and is, of course, invariant under rigid motions. It turned out to be so robust, that the tolerance parameter for the voting mechanism was set to only 1.0. First attempts to find the correct rotation indicated that too many false matches between points at nearly flat regions of the surfaces contributed false votes to the voting tables. To fix this, we have excluded from the generic voting list all the points having footprints in the range 170°–190°. This improved the matching, and also reduced significantly the running time, due to data reduction. The algorithm found the rotation (6, 4.9, 180.9) and the shift (16, 41, 0). Figs. 4e, 4f, and 4g show isometric, top, and side views, respectively, of the first scan imposed on the second after applying this transformation. The three figures show that the two surfaces match quite closely in their overlapping portions. Fig. 5 shows typical distributions of votes in the voting tables. Fig. 5a shows the voting table that corresponds to the zero rotation, whereas Fig. 5b shows the voting table that corresponds to the rotation which the algorithm found to be correct. Every cell in the voting table is represented in the figures by a small ball whose radius is proportional to the number of votes in that cell. The distribution of the "bad" votes (the "cloud" of votes) in Figs. 5a and 5b is not uniform, and is, in a certain sense, a "biased" version of the Minkowski difference between the two images, after applying the rotation to the first image, where points in one image are subtracted from all points in the other image with similar footprints. The accumulation point of the "good" votes, which reflects the correct translation, is clearly visible in Fig. 5b. In "astrophysical" terms, this cluster is a "black hole," representing about 16 percent of the votes.)

In the next example, we looked again for a partial surface match between two halves of a model of an aircraft. Figs. 6a and 6b show the full aircraft in a wire-frame and in a shaded display, respectively. Figs. 6c and 6d show, respectively, the right and left halves of the aircraft, as seen from the front of the aircraft. The data consisted of two polyhedral surfaces containing 1,336 and 1,157 vertices, which we chose as the representative points. The footprints chosen for this example aim again to encode the curvature of the surface. This time

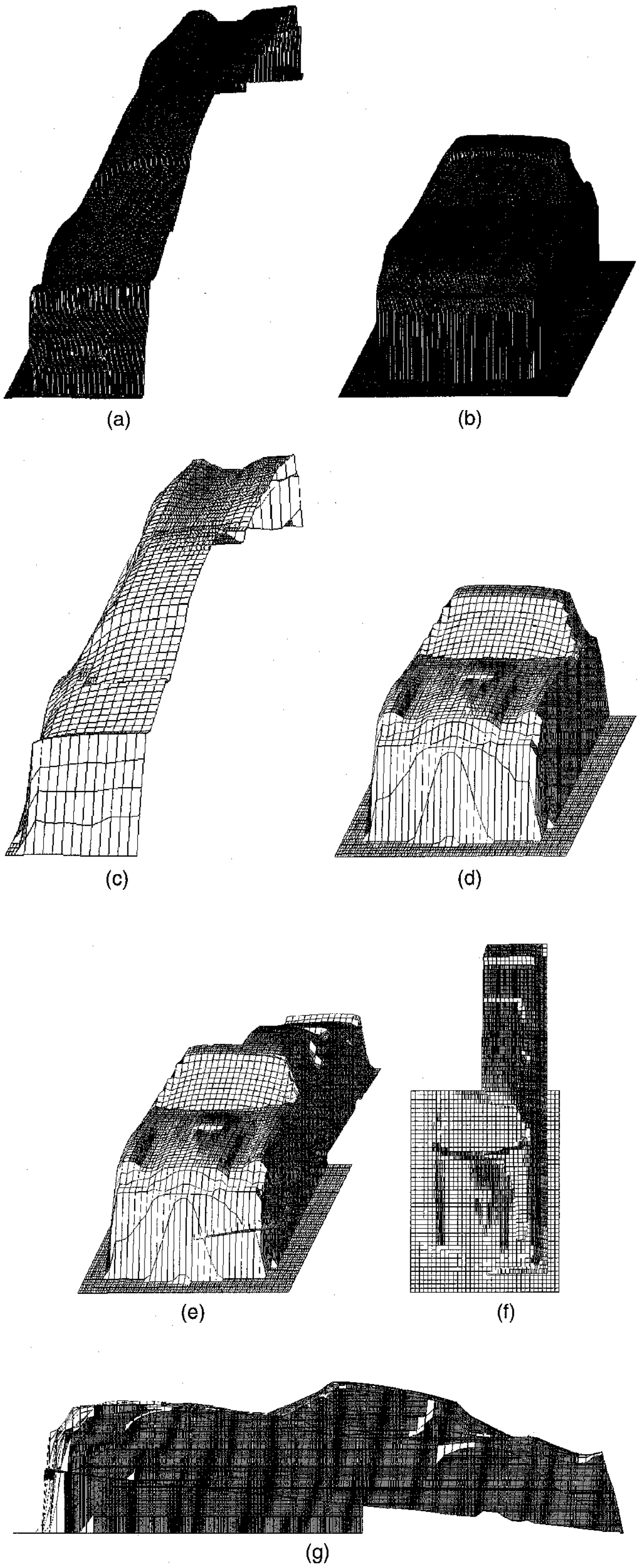
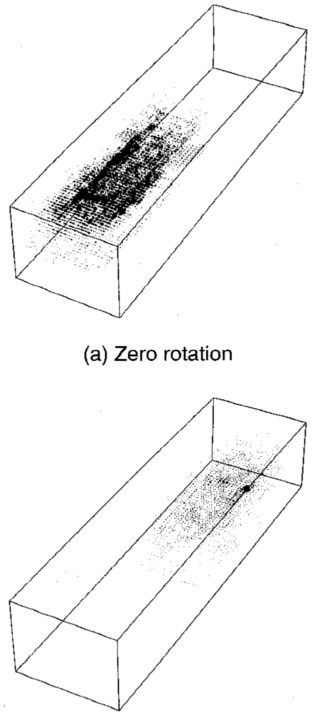


Fig. 4. A partial surface matching of digitized objects.



(a) Zero rotation

(b) Correct rotation

Fig. 5. Three-dimensional voting tables of the car.

we averaged for each vertex the dihedral angles between all pairs of neighboring facets incident to the vertex. We weighed each dihedral angle proportionally to the length of the edge common to the two facets. This footprint is again, of course, invariant under rigid motions. As with the previous example, we have excluded from the generic voting list all the points having footprints in the range 170°–190°. The algorithm found the rotation (–1.5, –1, 0.25), which received an overwhelmingly large score. The corresponding voting table contained four huge clusters, whose center cells received 93, 93, 76, and 73 votes, respectively. (The next "popular" cell received only 15 votes.) These cells corresponded to the translations (0, 32, 0), (0, 40, 0), (45, 36, 8), and (–45, 36, –8). Figs. 6e, 6f, 6g, and 6h show the first half of the aircraft imposed on the second half after applying these translations. It is seen clearly in all the four figures that two engines, one of each half of the aircraft, were matched almost perfectly. Fig. 7 shows the voting table that corresponds to the found rotation. The four accumulation points and their corresponding disks of votes are clearly visible in the figure.

Next we looked for the axis of symmetry of the CAD object shown in Figs. 8a and 8b (a wire-frame and a shaded display, respectively). We applied the same procedure as for the aircraft example in order to choose the representative points and their footprints. As expected, the initial phase gave the highest score to the zero rotation and translation, so we passed to the advancing phases the rotation that received the second highest score. The algorithm found the relative rotation (180, 0, 0) (between the object and itself) with the respective translation (0, 0, 0). This solution means that the object has a π-symmetry about the x-axis.

The next several examples are from molecular biology. The first two of them consist of the same input. The data describes two subunits (chains) of human deoxyhemoglobin (a protein found inside red cells of the human blood), having the well-known α-β dimer [32]. This dimer is reflected by a

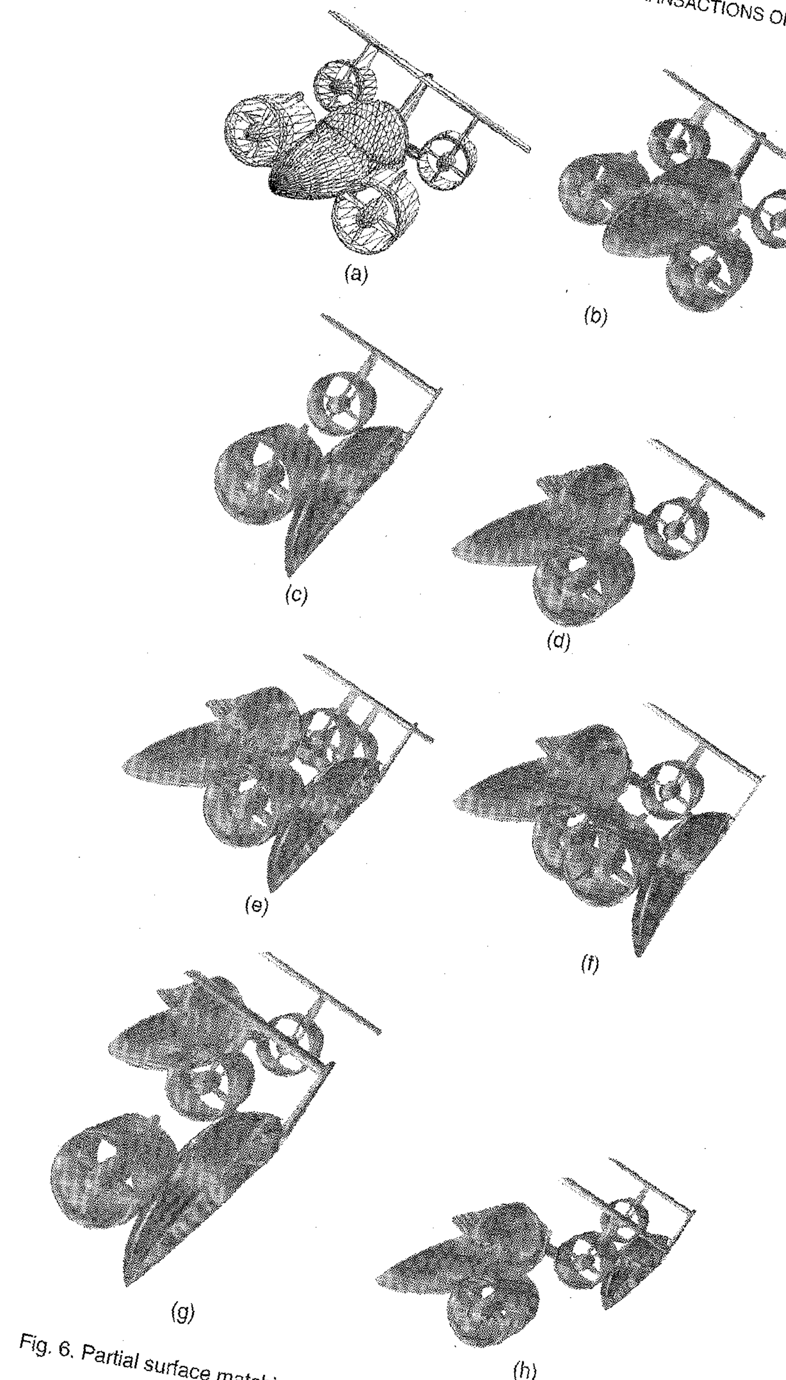


Fig. 6. Partial surface matchings of models of an aircraft.

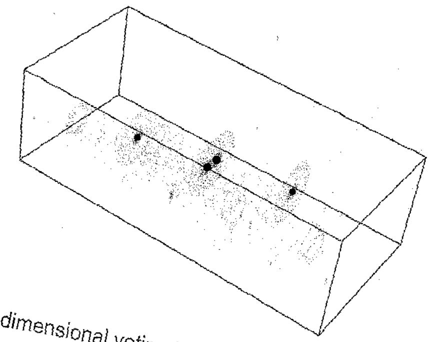


Fig. 7. A three-dimensional voting table of the aircraft.

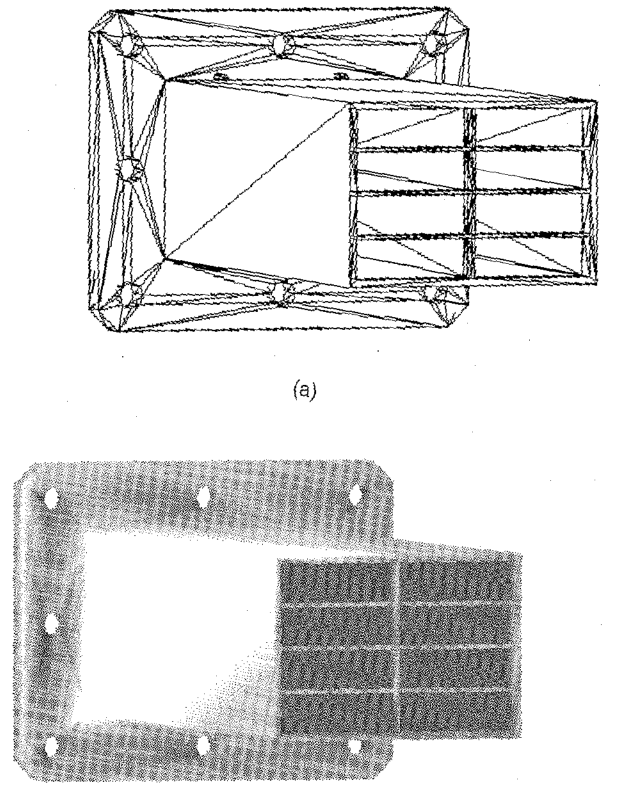


Fig. 8. A symmetric CAD model.

geometric match between portions of the boundaries of the two subunits, with volume complementarity near the matched boundaries. The data was obtained from the Brookhaven Protein Data Bank, where it is denoted by 2HHB. Each molecule is given as a list of atoms and the spatial coordinates of their centers. Fig. 9a shows the two subunits (the α chain is darker), already in the docking position.

As was mentioned in the introduction, previous methods for matching (docking) molecules usually applied a preprocessing step, which attempts to locate the "boundary" of the molecules, and to discard all the "inner" atoms which do not play a role in the matching. We invoked a simple procedure which (not quite accurately) tried to achieve a similar effect. For each atom a, We checked the eight axis-parallel octants with apex a, within L_∞-distance of 5Å from a. If every one of the eight limited octants intersected at least one other atom, we considered a to be "totally inner," and discarded it. We computed the footprint of each remaining "boundary" atom as the "amount of material" (including inner atoms) found inside a $9 \times 9 \times 9$ cube centered at the atom. Specifically, we computed how many of the 729 voxels in this cube contain the center of at least one atom. ("Volumes" of atoms connected by van der Waals bonds overlap.) Since we are interested here in matches that have volume complementarity, we summed up the footprints of every candidate pair of atoms, and if the result was sufficiently close to 729, then the pair of atoms was added to the generic voting list. The data was

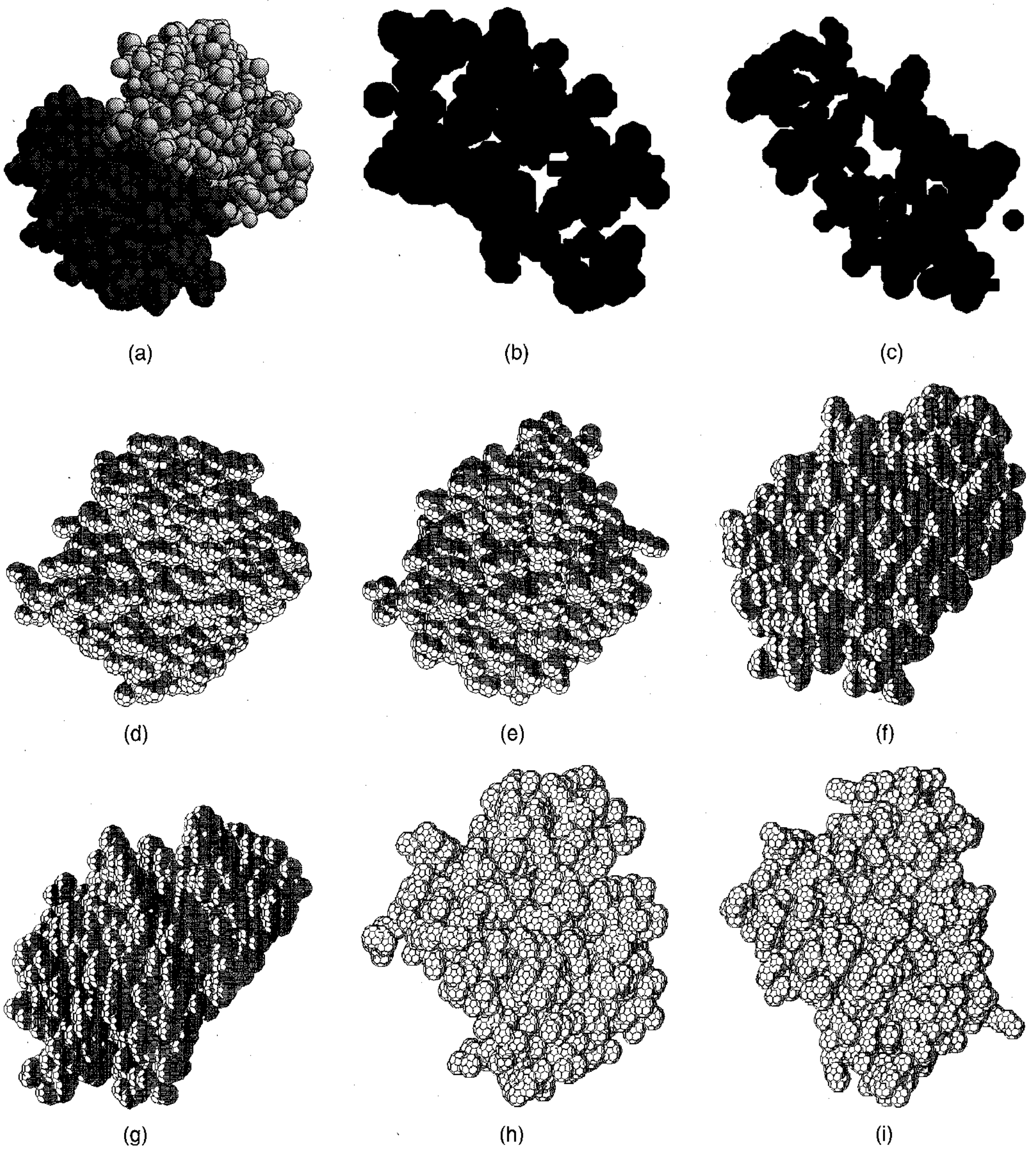


Fig. 9. Full volume and partial surface matching of hemoglobin subunits.

given so that the correct solution that reflects the contact between the α and the β chains should have been the zero rotation and translation. The algorithm found the best rotation to be (–0.25, 0, 0). The simple technique for finding the final translation failed, since the voting table of the correct rotation was still too sparse, and no distinguished accumulation point could be identified. We feel that this happened since we did not compute accurately enough the boundary surfaces of the molecule subunits, and because the chosen footprints were not descriptive enough. We believe, however, that the tolerance parameter for the voting mechanism was accurate. Given the correct rotation, we applied a simple correlation function similar to that described in [55] and in Section 4.4. We set $c_A = -15$ and $c_B = 2$. The program checked a 5Å-grid of translations in the range of –30Å to 30Å in each direction, and found the correct translation to be (–2, 0, 1).

In order to verify the docking, we computed the intersection of the two molecules with a series of parallel planes. Figs. 9b and 9c show the intersections with the plane $X = 14.7$ (at the original world coordinates). The matching area is seen between the upper right side of Fig. 9b and the lower left side of Fig. 9c. These figures repeat the results presented at [55]. Next, we looked for a volume match between these two subunits of hemoglobin. The shapes of the two chains resembled each other, and indeed, the algorithm found a rotation and a translation, which made them almost fully overlap. Figs. 9d and 9e show the two subunits, side by side, after applying the rotation. The two subunits are shown in Figs. 9f and 9g and in Figs. 9h and 9i in the same orientation but from different view points. These overlaps reveal structural motifs which are common to the two chains. Indeed, the α and the β chains are very similar to each other in their tertiary structure, which consists of similar lengths of an α-helix with bends of about the same angles and directions [65, p. 145].

We applied the same procedure in order to identify a

TABLE 2
PERFORMANCE OF THE MATCHING ALGORITHM ON MOLECULE DOCKING

Case	Figure	|*A*|	|*B*|	Tolerance Parameter	Voting Pairs
2HHB	9	1,112[1]	1,166[1]	30	560
2MHB	10	1,113[2]	1,178[2]	30	357
4MBN		44[3]	1,339[3]	30	823

Case	# of Initial Starting Points	Computed Rotation		Computed Translation
		Phase II	Phase III	
2HHB	1	(0.0, 0.0, 0.0)	(–0.25, 0.0, 0.0)	(-2,0,1)[4]
2MHB	5	(2.0, 0.0, 2.0)	(1.875, 0.0, 0.0)	(1, 0, –1)[4]
4MBN	2	(–2.0, 0.0, –2.0)	(–1.688, 0.562, –1.75)	(–3, –1, 4)

Case	# of Scoring Operations				Time (Sec.)				
	I	II	III	Total	I	II	III	Total	Per S.O.
2HHB	108	54	126	288	80	40	94	214	0.74
2MHB	108	162	450	720	73	112	310	495	0.68
4MBN	108	162	324	594	88	134	273	495	0.83

[1] *473 and 480, respectively, after eliminating "inner" points.*
[2] *479 and 498, respectively, after eliminating "inner" points.*
[3] *35 and 553, respectively, after eliminating "inner" points.*
[4] *Translation computed by the correlation technique; its timing is not included here.*

similar α-β dimer of two subunits of horse methemoglobin (PDB code 2MHB). Here the areas of the matching boundary portions were smaller than those in the 2HHB case. Since the initial phase (scoring a coarse grid of points in the space of rotations) gave high scores to several rotations, we passed to the advancing phases the five rotations that received the highest scores. The algorithm identified two significant maxima of the scoring function: the rotations (–1.688, 5.312, 174) and (1.875, 0, 0). The latter rotation represents the contact between the α and the β chains. The correlation function, in its setting for the 2HHB case, did not score high enough the correct translation for the contact. We modified the cell-to-cell contribution to be three in case the two cells were on the subunit boundaries, –6 in case one cell was on the boundary of one subunit and the other cell was internal to the second subunit, and –10 in case the two cells were internal to the subunits. These values were found empirically. This version of the correlation function found the correct translation within a very small error. Fig. 10 shows the docking of the two methemoglobin subunits in the contact between them. We have also attempted to dock a receptor (myoglobin) with a ligand (heme). For this purpose we used myoglobin from a sperm whale (PDB code 4MBN). See Table 2 for the full details.

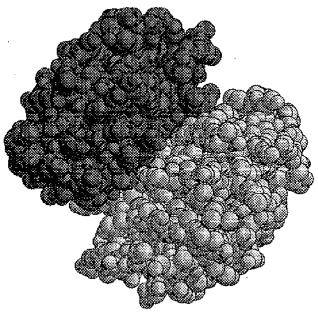


Fig. 10. Docking of horse methemoglobin subunits.

Finally, we applied our algorithm to the problem of registration of medical images. We compared three bit-volumes obtained by scanning a functional brain phantom[2] with an MRI scanner. The first two scans differed in their resolutions: the distance between consecutive slices in the first scan was 6 mm, whereas it was only 3 mm in the second scan. The phantom had the same orientation in the first two scans. The resolution of the third scan was as high as the second, but the phantom was rotated by –8 degrees around the x-axis, relative to the first two scans.

The data obtained by the scanner were three bit-volumes, where the dimensions of each voxel were $0.86 \times 0.86 \times 0.86$ mm^3. We faced a data-explosion problem in this experiment, too. The original bit-volumes contained 35,457, 35,898, and 36,111 points. We used only those points whose coordinates in the xyz-grid were integer multiples of five. Thus, the size of the data that we considered was reduced by a factor of roughly 125. Figs. 11a and 11b show the first and the third scans of the phantom from a side view point. (The second scan looks much the same as the first one.) As for the Geneva experiment, the footprints chosen for this example count the "amount of material" around each voxel. We considered only boundary voxels of the two-dimensional MRI scans (obtained by standard gray-level thresholding), and the footprint of a voxel was set to the number of voxels belonging to the phantom within the $5 \times 5 \times 5$ cube centered at the given voxel.

The tolerance parameter for the voting mechanism was set to one. We experimented with the three possible registrations

[2] A functional phantom is a piece of material (perspex, in our case), in which the shape of the organ is engraved. In order to simulate the scanning of the organ in different imaging modalities, special types of liquid (each modality with its own appropriate liquid) fill the engraved shape of the organ. Every imaging modality can be tuned to be sensitive to the corresponding special liquid, and almost insensitive at all to the phantom itself.

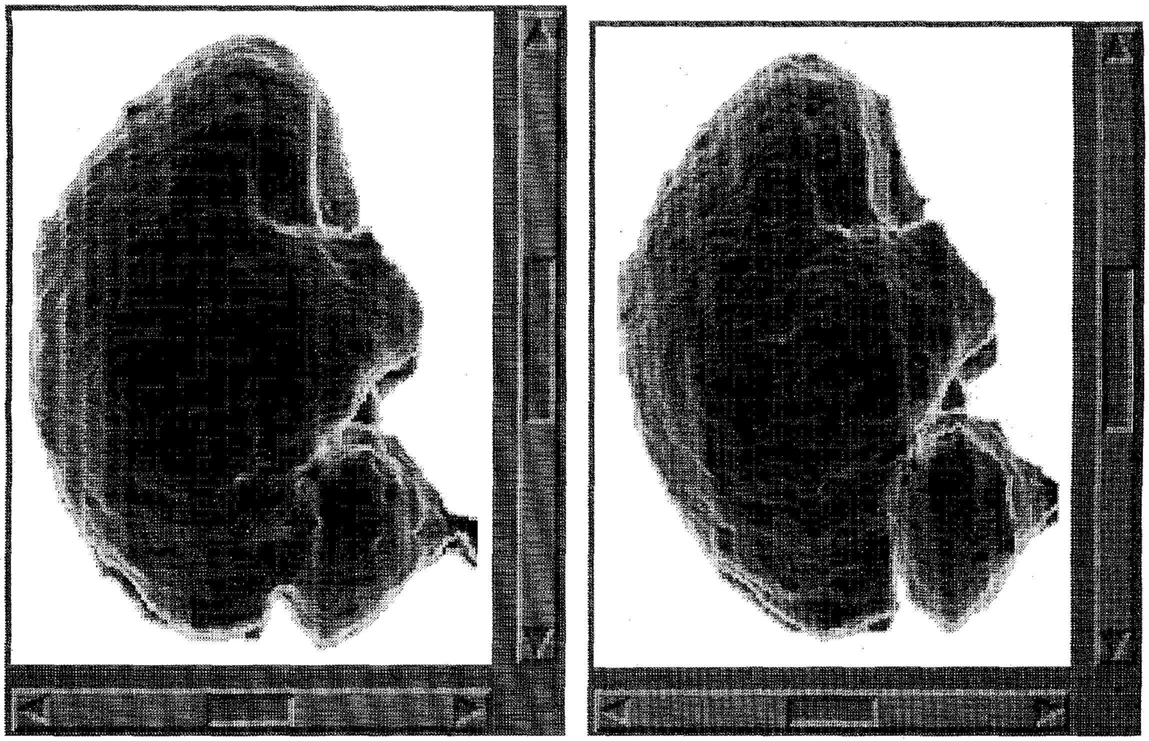


Fig. 11. Surface matching of a functional brain phantom. (a) Original. (b) Tilted by –8 degrees.

of pairs of scans (out of the three). The algorithm found the correct rotation and translation in all the three experiments with practically no error at all.

Table 1 summarizes the performance of our implementation on all the examples described above, except for the molecule docking. The latter experiments are summarized in Table 2. In all the docking experiments we sought a surface matching, the footprint was the same as for the volume matching of the molecules (see Table 1), and the correct rotation and translation were the identity. All the time measurements in Tables 1 and 2 were taken on an SGI Indigo workstation. Our experimental results show that the time needed for a table-scoring operation depended primarily on the number of voting pairs and on the average density of the voting tables, which depends more on the particular matching instance and less on the specific rotation. This was reflected well in the molecule docking experiments, in which a scoring operation usually required more time than in the other experiments, due to the larger sparseness of the voting tables in this case. The total running time for a specific instance depended roughly linearly on the number of scoring operations, since the difference (in running time) between different scoring operations for the same data was not significant. Typically, the third phase of the algorithm required significantly more time than the second phase (except for the Geneva case). In practice, however, the third phase could be omitted, because the results of the second phase were already reasonably accurate. Each of our experiments took a few minutes to run. However, we can trade time for accuracy, by reducing the tolerance parameter (thereby reducing the number of voting pairs), by reducing the resolution of the xyz-grid (thereby making the voting tables denser), and/or by reducing the resolution of the search in the three-dimensional space of rotations (thereby reducing the number of scoring operations).

We have also implemented the statistical method described in Section 5 and applied it to the matching of the snapshots of the car. The principal components analysis revealed the characteristic values 170.54, 24.53, and 8.40, with the corresponding characteristic vectors (0.012, 0.999, –0.045), (0.994, –0.008, 0.105), and (–0.105, 0.046, 0.993), respectively. The third characteristic vector was thus in the direction of the axis of rotation ℓ. We then projected the two data sets on a plane orthogonal to ℓ, and computed the relative rotation between the projected sets according to the method described in Section 5 (the resulting angle was 179.0125). Putting everything together, we got that the rotational component of the correct transformation was

$$\begin{pmatrix} -0.978 & -0.024 & -0.207 \\ 0.010 & -0.997 & 0.071 \\ -0.208 & 0.067 & 0.976 \end{pmatrix}$$

(All the computations were performed in a much higher precision than the one shown here.) This compares to the rotation matrix

$$\begin{pmatrix} -0.996 & 0.006 & -0.086 \\ -0.015 & -0.995 & 0.103 \\ -0.085 & 0.104 & 0.991 \end{pmatrix}$$

that we got with the regular method. We did not need to further advance from this rotation to better rotations. The running time on an SGI Indigo workstation was the following: the principal components analysis required two seconds, projecting the data onto a two-dimensional problem was negligible, solving the two-dimensional problem required 16 seconds, and obtaining the final rotational component of the solution was again negligible. In total, the statistical approach required 18 seconds in this case, compared with 188 seconds for the regular method. We did not achieve adequate results with the statistical method in any other experiment.

8 Conclusion

We have proposed in this paper an algorithm for solving the practical problem of partial surface or volume matching between two objects in three-space. This problem is a basic and important problem in pattern recognition and computer vision, with many industrial, chemical, and medical applications. We have treated separately the rotation and the translation components of the relative rigid motion between the two objects. We developed a two-step technique (voting followed by scoring the vote) for measuring the goodness of a given rotation, then used this technique as a subroutine called by an iterative process that advances toward the best rotation. Finally, we computed the best translation between the objects, corresponding to the best rotation. We used footprints which counted the "amount of material" for volume matching, and footprints that approximate the surface curvature for surface matching. However, any other "descriptive" footprint system might serve as well. As demonstrated, our technique found accurately the matches between several pairs of objects, taken from several totally different domains. A manual inspection of the resulting voting tables shows that they are fairly noisy. The typical table looks like a cloud of randomly and sparsely (though generally not uniformly) spread points, containing a small slightly denser region where the good votes are clustered. One of the open problems that we pose for future research is to develop alternative theoretically-sound and practically-efficient methods for identifying such clusters of data in otherwise randomly spread data.

There are several research directions that we plan to pursue and to explore further. We plan to continue the experimentation with our algorithm, to test its performance limits and see if there are data instances on which the algorithm needs further fine-tuning. We also plan to explore several enhancements and improvements of the algorithm. These include the design of better footprints (especially in the molecular biology domain), experimentation with other scoring functions, further study of statistical approaches, design of better searching mechanisms, etc. In a companion work, we are presently investigating the use of directed footprints, e.g., footprints which include the normals to the surface. This component is not invariant under rotation, but may have a significant role in the quality of the footprint system. We plan to improve the implementation of the voting parameters in the molecule docking problem. Specifically, we plan to improve the determination of the atoms on the boundaries of the molecules, and to try to compute better footprints for them. Moreover, we plan to enhance the footprints, so that they reflect not only the geometries of the molecules, but also chemical properties of the atoms, such as their bases, electrical potentials, possible bondings, etc., which also play a significant role in the chemical reaction between molecules.

Acknowledgments

Work on this paper by the authors has been supported by the G.I.F.—the German-Israeli Foundation for Scientific Research and Development. Work by the first author has been supported also by the Israeli Ministry of Science and the Arts under an Eshkol Grant 0562-1-94. Work by the second author has been supported also by National Science Foundation Grants CCR-91-22103 and CCR-93-11127, by grants from the U.S.-Israeli Binational Science Foundation, and by the Fund for Basic Research administered by the Israeli Academy of Sciences.

We wish also to thank Haim Wolfson for helpful discussions concerning the molecule docking, and David Steinberg for his advice concerning statistical theory and techniques. The CAD data were contributed by Cubital Ltd. The digitization files of the car were supplied by Sharnoa Ltd. The molecule descriptions were obtained from the Brookhaven Protein Data Bank. The brain phantom data was supplied by Algotec Systems Ltd.

References

[1] R.A. Abagyan and N.V. Maiorov, "A Simple Qualitative Representation of Polypeptide Chain Folds: Comparison of Protein Tertiary Structures," *J. Biomolecular Structure and Dynamics*, vol. 5, pp. 1,267-1,279, 1988.

[2] T. Abel and T. Maniatis, "Gene Regulation: Action of Leucine Zippers," *Nature*, vol. 341, pp. 24-25, 1989.

[3] N.M. Alpert, J.F. Bradshaw, D. Kennedy, and J.A. Correia, "The Principal Axes Transformation—A Method for Image Registration," *J. Nuclear Medicine*, vol. 31, pp. 1,717-1,722, 1990.

[4] T.W. Anderson, *Introduction to Multivariate Statistical Analysis*. John Wiley & Sons, Inc., 1958.

[5] K.S. Arun, T.S. Huang, and S.D. Blostein, "Least Square Fitting of Two 3-D Point Sets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698-700, 1987.

[6] R. Bajcsy and S. Kovacic, "Multiresolution Elastic Matching," *Computer Vision, Graphics, and Image Processing*, vol. 46, pp. 1-21, 1989.

[7] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, "Parametric Correspondence and Chamfer Matching: Two New Techniques for Image Matching," *Proc. Fifth Int'l Joint Conf. Artificial Intelligence*, pp. 659-663, 1977.

[8] F.P. Beer and E.R. Johnston, Jr., *Vector Mechanics for Engineers: Dynamics*. McGraw-Hill, 1986.

[9] P.J. Besl, "Geometric Modeling and Computer Vision," *Proc. IEEE*, vol. 76, pp. 936-958, 1988.

[10] P.J. Besl, "The Free-Form Surface Matching Problem," *Machine Vision for Three-Dimensional Scenes*, H. Freeman, ed. New York: Academic, 1990.

[11] P.J. Besl and R.C. Jain, "Three-Dimensional Object Recognition," *ACM Computing Surveys*, vol. 17, no. 1, pp. 75-154, Mar. 1985.

[12] P.J. Besl and N.D. McKay, "A Method for Registration of 3D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[13] A. Blake and A. Yuille, *Active Vision*. Cambridge, Mass.: MIT Press, 1993.

[14] R.C. Bolles and P. Horaud, "3DPO: Three-Dimensional Part Orientation System," *Int'l J. Robotics Research*, vol. 5, pp. 3-26, 1986.

[15] F.L. Bookstein, "From Medical Images to the Biometrics of Form," *Information Processing in Medical Imaging*, S.L. Bacharach, ed. Dor-

drecht, The Netherlands: Martinus Nijhoff Publishers, 1986. pp. 1-18.

[16] F.L. Bookstein, "Thin-Plate Splines and the Atlas Problem for Biomedical Images," *Information Processing in Medical Imaging*, A.C.F. Colchester and D.J. Hawkes, eds. Berlin: Springer-Verlag, 1991, pp. 326-342.

[17] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 849-865, 1988.

[18] P. Brou, "Using the Gaussian Image to Find the Orientation of an Object," *Int'l J. Robotics Research*, vol. 3, pp. 89-125, 1983.

[19] L.G. Brown, "A Survey of Image Registration Techniques," *ACM Computing Surveys*, vol. 24, pp. 325-376, 1992.

[20] E. de Castro, G. Cristini, A. Martelli, C. Morandi, and M. Vascotto, "Compensation of Random Eye Motion in Television Ophthalmoscopy: Preliminary Results," *IEEE Trans. Medical Imaging*, vol. 6, pp. 74-81, 1987.

[21] B. Chazelle, "A Functional Approach to Data Structures and Its Use in Multidimensional Searching," *SIAM J. Computing*, vol. 17, pp. 427-462, 1988.

[22] C.T. Chen, C.A. Pelizzari, G.T.Y. Chen, M.D. Cooper, and D.N. Levin, "Image Analysis of PET Data With the Aid of CT and MR Images," *Information Processing in Medical Imaging*, C.N. de Graaf and M.A. Viergever, eds. New York: Plenum Press, 1988, pp. 601-611.

[23] R.T. Chin and C.R. Dyer, "Model-Based Recognition in Robot Vision," *ACM Computing Surveys*, vol. 18, no. 1, pp. 67-108, Mar. 1986.

[24] D.L. Collins, T.M. Peters, W. Dai, and A.C. Evans, "Model-Based Segmentation of Individual Brain Structures From MRI Data," *Proc. SPIE Visualization in Biomedical Computing*, R.A. Robb, ed. Bellingham, Wash.: SPIE Press, 1992, pp. 10-23.

[25] M.L. Connolly, "Shape Complementarity at the Hemoglobin α_1-β_1 Subunit Interface," *Biopolymers*, vol. 25, pp. 1,229-1,247, 1986.

[26] P.A. van der Elsen, J.B.A. Maintz, and M.A. Viergever, "Geometry Driven Multimodality Image Matching," *Brain Topography*, vol. 5, no. 2, pp. 153-158, 1992.

[27] P.A. van der Elsen, E.J.D. Pol, and M.A. Viergever, "Medical Image Matching—A Review With Classification," *IEEE Eng. in Medicine and Biology*, vol. 12, no. 1, pp. 26-39, 1993.

[28] A.C. Evans, S. Marrett, L. Collins, and T.M. Peters, "Anatomical-Functional Correlative Analysis of the Human Brain Using Three Dimensional Imaging Systems," *Proc. SPIE Medical Imaging III: Image Processing*, R.H. Schneider, S.J. Dwyer III, and J.R. Gilbert, eds. Bellingham, Wash.: SPIE Press, 1989, pp. 264-274.

[29] T.J. Fang, Z.H. Huang, L.N. Kanal, B. Lambird, D. Lavine, G. Stockman, and F.L. Xiong, "Three-Dimensional Object Recognition Using a Transformation Clustering Technique," *Proc. Sixth IAPR and IEEE Int'l Conf. Pattern Recognition*, 1982, pp. 678-681.

[30] O.D. Faugeras, "New Steps Toward a Flexible 3D Vision System for Robotics," *Proc. IEEE Seventh Int'l Conf. Pattern Recognition*, Montreal, Canada, July 30-Aug. 2, 1984, pp. 796-805.

[31] O.D. Faugeras and M. Hebert, "A 3D Recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces," *Proc. Seventh Int'l Joint Conf. Artificial Intelligence*, Vancouver, B.C., Canada, Aug. 24-28, 1983, pp. 996-1,002.

[32] G. Fermi, M.F. Perutz, B. Shaanan, and R. Fourme, "The Crystal Structure of Human Deoxyhemoglobin at 1.74Å Resolution," *J. Molecular Biology*, vol. 175, pp. 159-174, 1984.

[33] D. Fischer, O. Bachar, R. Nussinov, and H.J. Wolfson, "An Efficient Computer Vision Based Technique for Detection of Three Dimensional Structural Motifs in Proteins," *J. Biomolecular Structure and Dynamics*, vol. 9, pp. 769-789, 1992.

[34] D. Fischer, R. Norel, R. Nussinov, and H.J. Wolfson, "3D Docking of Protein Molecules," *Proc. Fourth Symp. Combinatorial Pattern Matching, Lecture Notes Computer Science*, 684. Berlin: Springer Verlag, 1993, pp. 20-34.

[35] D. Fischer, R. Nussinov, and H.J. Wolfson, "3D Substructure Matching Protein Molecules," *Proc. Third Symp. Combinatorial Pattern Matching, Lecture Notes in Computer Science*, 644. Berlin: Springer Verlag, 1992, pp. 136-150.

[36] R.B. Fisher, "Using Surfaces and Object Models to Recognize Partially Obscured Objects," *Proc. Eighth Int'l Joint Conf. Artificial Intelligence*, Karlsruhe, West Germany, Aug. 8-12, 1983.

[37] A. Gamboa-Aldeco, L.L. Fellingham, and G.T.Y. Chen, "Correlation of 3D Surfaces From Multiple Modalities in Medical Imaging," *Proc. SPIE Medicine XIV/PACS IV*, R.H. Schneider and S.J. Dwyer III, eds. Bellingham, Wash.: SPIE Press, 1986, pp. 467-473.

[38] G. Garibotto, C. Giorgi, and U. Cerchiari, "3D Image Processing in Functional Stereotactic Neurosurgery," *Proc. SPIE Applications of Digital Image Processing*. Bellingham, Wash: SPIE Press, 1983, pp. 280-287.

[39] H. Goldstein, *Classical Mechanics*. Reading, Mass.: Addison-Wesley, 1980.

[40] G.H. Golub and C.F. van Loan, *Matrix Computations*. Baltimore, Md.: Johns Hopkins Univ. Press, 1983.

[41] D.J. Hawkes, D.L.G. Hill, E.D. Lehmann, G.P. Robinson, M.N. Maisay, and A.C.F. Colchester, "Preliminary Work on the Interpretation of SPECT Images With the Aid of Registered MR Images and an MR Derived 3D Neuro-Anatomical Atlas," *3D Imaging in Medicine*, K.H. Höhne, H. Fuchs, and S.M. Pizer, eds. Berlin: Springer-Verlag, 1990, pp. 241-251.

[42] D.R. Haynor, A.W. Borning, B.A. Griffin, J.P. Jacky, I.J. Kalet, and W.P. Shuman, "Radiotherapy Planning: Direct Tumor Location on Simulation and Port Films Using CT," *Radiology*, vol. 158, no. 2, pp. 537-540, 1986.

[43] B.K.P. Horn, "Extended Gaussian Images," *Proc. IEEE*, vol. 72, pp. 1,656-1,678, Dec. 1984.

[44] B.K.P. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *J. Opt. Soc. Amer.*, vol. 4, pp. 629-642, 1987.

[45] D.L.G. Hill, D.J. Hawkes, J.E. Crossman, M.J. Gleeson, T.C.S. Cox, E.C.M.L. Bracey, A.J. Strong, and P. Graves, "Registration of MR and CT Images for Skull Base Surgery Using Point-Like Anatomical Features," *British J. Radiology*, vol. 64, pp. 1,030-1,035, 1991.

[46] D.L.G. Hill, D.J. Hawkes, and C.R. Hardingham, "The Use of Anatomical Knowledge to Register 3D Blood Vessel Data Derived From DSA With MR Images," *Proc. SPIE Image Processing*. Bellingham, Wash.: SPIE Press, 1991, pp. 348-357.

[47] J. Hong and H.J. Wolfson, "An Improved Model-Based Matching Method Using Footprints," *Proc. Ninth Int'l Conf. Pattern Recognition*, Rome, Nov. 1988, pp. 72-78.

[48] D.P. Huttenlocher, J.J. Noh, and W.J. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes," *Proc. IEEE Int'l Conf. Computer Vision*, 1993.

[49] D.P. Huttenlocher and S. Ullman, "Recognizing Solid Objects by Alignment With an Image," *Int'l J. Computer Vision*, vol. 5, pp. 195-212, 1990.

[50] H. Jiang, K. Holton, and R. Robb, "Image Registration of Multimodality 3D Medical Images by Chamfer Matching," *Proc. SPIE Biomedical Image Processing and Three-Dimensional Microscopy*. Bellingham, Wash.: SPIE Press, 1992, pp. 356-366.

[51] F. Jiang and S.H. Kim, "Soft Docking: Matching of Molecular Surface Cubes," *J. Molecular Biology*, vol. 219, pp. 79-102, 1991.

[52] L. Junck, J.G. Moen, G.D. Hutchins, M.B. Brown, and D.E. Kuhl, "Correlation Methods for the Centering, Rotation, and Alignment of Functional Brain Images," *J. Nuclear Medicine*, vol. 31, pp. 1,220-1,226, 1990.

[53] A. Kalvin, E. Schonberg, J.T. Schwartz, and M. Sharir, "Two-Dimensional, Model-Based, Boundary Matching Using Footprints," *Int'l J. Robotics Research*, vol. 5, pp. 38-55, 1986.

[54] B. Kamgar-Parsi, J.L. Jones, and A. Rosenfeld, "Registration of Multiple Overlapping Range Images: Scenes Without Distinctive Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1989.

[55] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A.A. Friesem, C. Aflalo, and I.A. Vakser, "Molecular Surface Recognition: Determination of Geometric Fit Between Proteins and Their Ligands by Correlation Techniques," *Proc. National Academy of Sciences of the USA (Biophysics)*, vol. 89, pp. 2,195-2,199, Mar. 1992.

[56] P.A. Kenny, D.J. Dowsett, D. Vernon, and J.T. Ennis, "A Technique for Digital Image Registration Used Prior to Subtraction of Lung Images in Nuclear Medicine," *Physical Medical Biology*, vol. 35, pp. 679-685, 1990.

[57] E. Kishon, T. Hastie, and H. Wolfson, "3D Curve Matching Using Splines," *J. Robotic Systems*, vol. 8, pp. 723-743, 1991.

[58] S. Kovacic, J.C. Gee, W.S.L. Ching, M. Reivich, and R. Bajcsy, "Three-Dimensional Registration of PET and CT Images," *Proc. 11th Annual IEEE Int'l Conf. Engineering in Medicine and Biology*. Los Alamitos, Calif.: IEEE CS Press, 1989, pp. 548-549.

[59] F.S. Kuhl, G.M. Crippen, and D.K. Friesen, "A Combinatorial Algorithm for Calculating Ligand Binding," *Computational Chemistry*, vol. 5, pp. 24-34, 1984.

[60] I.D. Kuntz, J.M. Blaney, S.J. Oatley, R. Langridge, and T.E. Ferrin, "A Geometric Approach to Macromolecule-Ligand Interactions," *J. Molecular Biology*, vol. 161, pp. 269-288, 1982.

[61] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson, "On Recognition of 3D Objects From 2D Images," *Proc. IEEE Int'l Conf. Robotics and Automation*, pp. 1,407-1,413, 1988.

[62] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson, "Affine Invariant Model-Based Object Recognition," *IEEE Trans. Robotics and Automation*, vol. 6, pp. 578-589, 1990.

[63] Y. Lamdan and H.J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc IEEE Int'l Conf. Computer Vision*, pp. 238-249, 1988.

[64] S. Lavallee and R. Szeliski, "Recovering the Position and Orientation of Free-Form Objects From Image Contours Using 3D Distance Maps," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 378-390, 1995.

[65] A.L. Lehninger, *Biochemistry*. New York: Worth Publishers, Inc., 1978.

[66] V.R. Mandava, J.M. Fitzpatrick, and D.R. Pickens III, "Adaptive Search Space Scaling in Digital Image Registration," *IEEE Trans. Medical Imaging*, vol. 8, no. 3, pp. 251-262, 1989.

[67] E.M. Mitchel, P.J. Artymiuk, D.W. Rice, and P. Willet, "Use of Techniques Derived From Graph Theory to Compare Secondary Structure Motifs in Proteins," *J. Molecular Biology*, vol. 212, pp. 151-166, 1989.

[68] M. Moshfeghi, "Elastic Matching of Multimodality Medical Images," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 271-282, 1991.

[69] R. Nussinov and H.J. Wolfson, "Efficient Detection of Three-Dimensional Structural Motifs in Biological Macromolecules by Computer Vision Techniques," *Proc. National Academy of Sciences of the USA (Biophysics)*, vol. 88, pp. 10,495-10,499, Dec. 1991.

[70] C.A. Pelizzari, G.T.Y. Chen, D.R. Spelbring, R.R. Weichselbaum, and C.T. Chen, "Accurate Three-Dimensional Registration of CT, PET, and/or MR Images of the Brain," *J. Computer Assisted Tomography*, vol. 13, pp. 20-26, 1989.

[71] M. Potmesil, "Generation of 3D Surface Descriptions From Images of Pattern-Illuminated Objects," *Proc IEEE Conf. Pattern Recognition and Image Processing*, pp. 553-559, 1979.

[72] M. Potmesil, "Generating Models of Solid Objects by Matching 3D Surface Segments," *Proc. Eighth Int'l Joint Conf. Artificial Intelligence*, pp. 1,089-1,093, Karlsruhe, West Germany, Aug. 8-12, 1983.

[73] F.M. Richards and C.E. Kundrot, "Identification of Structural Motifs From Protein Coordinate Data: Secondary Structure and First-Level Supersecondary Structure," *Protein Structures*, vol. 3, pp. 71-84, 1988.

[74] D. Sankoff and J.B. Kruskal, *Time Warps, String Edits and Macromolecules*. Reading, Mass.: Addison-Wesley, 1983.

[75] J.T. Schwartz and M. Sharir, "Identification of Partially Obscured Objects in Two and Three Dimensions by Matching Noisy Characteristic Curves," *Int'l J. Robotics Research*, vol. 6, no. 2, pp. 29-44, Summer 1987.

[76] M. Singh, W. Frei, T. Shibata, G.C. Huth, and N.E. Telfer, "A Digital Technique for Accurate Change Detection in Nuclear Medical Images—With Application to Myocardial Perfusion Studies Using Thallium-201," *IEEE Trans. Nuclear Sciences*, vol. 26, pp. 565-575, 1979.

[77] H.A. Spang III, "A Review of Minimization Techniques for Nonlinear Functions," *SIAM Review*, vol. 4, pp. 343-365, 1962.

[78] G. Stockman, "Object Recognition and Localization Via Pose Clustering," *Computer Vision, Graphics, and Image Processing*, vol. 40, pp. 361-387, 1987.

[79] G. Stockman and J.C. Esteva, "Use of Geometrical Constraints and Clustering to Determine 3D Object Pose," *Proc. Seventh IAPR and IEEE Int'l Conf. Pattern Recognition*, pp. 742-744, Montreal, Canada, July 30-Aug. 2, 1984.

[80] R. Szeliski, "Estimating Motion From Sparse Range Data Without Correspondence," *Proc. Second Int'l Conf. Computer Vision*, pp. 207-216, 1988.

[81] G. Taubin, "About Shape Descriptors and Shape Matching," Technical Report LEMS-57, Div. Eng., Brown Univ., Providence, R.I., 1989.

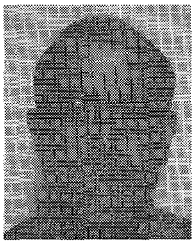
[82] K.D. Toennies, J.K. Udupa, G.T. Herman, I.L. Wornom III, and S.R. Buchman, "Registration of 3D Objects and Surfaces," *IEEE Computer Graphics & Applications*, vol. 10, no. 3, pp. 52-62, 1990.

[83] H.J. Wolfson, "Model-Based Object Recognition by Geometric Hashing," *Proc. First European Conf. Computer Vision, Lecture Notes in Computer Science*, 427. Berlin: Springer Verlag, 1990, pp. 526-536.

[84] H.J. Wolfson, "On Curve Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 483-489, May 1990.

Gill Barequet is currently a postdoctoral fellow at the Department of Computer Science of Johns Hopkins University. He received his BSc in mathematics and computer science, and his MSc and PhD in computer science from Tel Aviv University in 1985, 1987, and 1994, respectively.

His research interests include discrete and computational geometry, interpolation and reconstruction algorithms, geometric software and computing over the Internet, and geometric applications in medical imaging, molecular biology, and CAD. He holds five U.S. patents in related areas.



Micha Sharir is a Nizri Professor of Computational Geometry and Robotics in the Department of Computer Science at the School of Mathematical Sciences of Tel Aviv University. He received his PhD in mathematics from Tel Aviv University in 1976. His research interests include computational and combinatorial geometry and their applications, motion planning, and other topics in robotics.

He is the author of more than 200 research papers, and is a coauthor of *Davenport-Schinzel Sequences and Their Geometric Applications*. He is a visiting research professor at the Courant Institute of Mathematical Sciences, New York University, where he was also an associate director of the robotics lab.

He was awarded a Max-Planck Research Prize (jointly with Emo Welzl) in 1992, and is a director of the newly established Minerva Center in Geometry at Tel Aviv University.