Proc. Natl. Acad. Sci. USA
Vol. 88, pp. 10495–10499, December 1991
Biophysics

Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques

(three-dimensional structural comparison/crystallographic coordinates/efficient computer vision algorithm/macromolecular structure analysis)

RUTH NUSSINOV*† AND HAIM J. WOLFSON‡§

*Sackler Institute of Molecular Medicine, Faculty of Medicine and ‡Computer Science Department, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978 Israel; †Laboratory of Mathematical Biology, National Cancer Institute, National Institutes of Health, Frederick Cancer Research Facility, Building 469, Room 151, Frederick, MD 21702; and §Robotics Research Laboratory, Courant Institute of Mathematical Sciences, New York University, 715 Broadway, 12th Floor, New York, NY 10003

Communicated by Jacob T. Schwartz, July 29, 1991 (received for review February 1990)

ABSTRACT **Macromolecules carrying biological information often consist of independent modules containing recurring structural motifs. Detection of a specific structural motif within a protein (or DNA) aids in elucidating the role played by the protein (DNA element) and the mechanism of its operation. The number of crystallographically known structures at high resolution is increasing very rapidly. Yet, comparison of three-dimensional structures is a laborious time-consuming procedure that typically requires a manual phase. To date, there is no fast automated procedure for structural comparisons. We present an efficient $O(n^3)$ worst case time complexity algorithm for achieving such a goal (where n is the number of atoms in the examined structure). The method is truly three-dimensional, sequence-order-independent, and thus insensitive to gaps, insertions, or deletions. This algorithm is based on the geometric hashing paradigm, which was originally developed for object recognition problems in computer vision. It introduces an indexing approach based on transformation invariant representations and is especially geared toward efficient recognition of partial structures in rigid objects belonging to large data bases. This algorithm is suitable for quick scanning of structural data bases and will detect a recurring structural motif that is *a priori* unknown. The algorithm uses protein (or DNA) structures, atomic labels, and their three-dimensional coordinates. Additional information pertaining to the structure speeds the comparisons. The algorithm is straightforwardly parallelizable, and several versions of it for computer vision applications have been implemented on the massively parallel connection machine. A prototype version of the algorithm has been implemented and applied to the detection of substructures in proteins.**

One of the basic emerging principles in molecular biology is the modular nature of DNA sequence elements and of the corresponding sequence-specific protein factors recognizing them. The domains appear to be independent units (1). Structural and functional studies of these domains have demonstrated the existence of several structural motifs. The motifs include the helix–turn–helix (HTH) (2), zinc fingers (3), homeodomain (4), leucine zipper (5), helix–loop–helix (6), Ser-Pro-Lys-Lys histone (7), proline-rich (8) and glutamine-rich (9) motifs, the antiparallel β-sheet (10) apparently inserted in the minor groove, and more recently a pair of β-strands in the major groove of the DNA (11). All of these motifs typically include less than 100 amino acid residues. Finding a given structural motif in a protein may clearly aid in understanding its role (12). The latter is inferred by analogy with other proteins containing the motif. Structural comparisons are thus central to molecular biology. The problem we are faced with is to devise efficient techniques for routine scanning of structural data bases and searching for recurrences of inexact structural motifs. The degree of allowed errors is to be determined by the user.

The most commonly used computerized macromolecule comparison approaches deal mainly with comparison of the primary structure of molecules. They are based on character string comparison algorithms, most of which use variations of the dynamic programming technique (for a good survey, see ref. 13). Structural comparison is superior to this primary sequence analysis, since it takes into account the spatial geometric structure of the molecules involved and not only their order on the primary chain. The increasing need for direct structural analysis of macromolecules has led to the development of several computerized methods (14–16). These methods, however, look for predefined motifs in the secondary structure of the macromolecule. Moreover, these motifs are usually composed of contiguous amino acids on the primary chain, such as α-helices or β-sheets. The method that we develop enables elucidating similar substructures in different molecules without specifying in advance what these structures should be. Moreover, the motifs do not necessarily involve contiguous amino acids, so the approach is truly three dimensional (3D). This enables detection of various structural patterns.

Currently, true 3D structural comparisons are carried out mainly using interactive computer graphics and visualization facilities. The programs compare the locations of every pair of corresponding atoms in any two specific structures. Although useful, this tool falls short of what is needed. Since the computer graphic programs compare either two complete (crystal or computed) structures or any user-specified subsections, they are excellent for individual protein or nucleic acid analysis but are very time consuming for extensive comparisons.

From a mathematical standpoint, the structural comparison problem between two molecules can be formulated as follows. Given the 3D coordinates of the atoms of two molecules, find a rigid transformation (rotation and translation) in space so that a "large" number of atoms of one molecule matches the atoms of the other molecule. The matching should preserve not only the geometric constraints of a rigid body but also the "labeling" constraints of the individual atoms (i.e., atom types) and their relevant chemical links. Moreover, one needs an efficient comparison technique of each structure versus all previously known structures simultaneously.

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

Abbreviations: 2D and 3D, two and three dimensional, respectively; HTH, helix–turn–helix; RS, reference set.

The mathematical problem stated above is closely related to the model-based recognition problem of 3D rigid objects. This problem has been intensively investigated in computer vision. One of the major problems in this field is to discover previously known objects in scenes, where some of the objects might appear to partially occlude each other. This is the, so called, model-based object recognition task (for extensive surveys, see refs. 17 and 18). By considering a molecule as an object consisting of many rigidly connected features (atoms), one can apply some of the computer vision techniques to our problem. Partial occlusion here is equivalent to the absence of partial substructures.

Several techniques have been suggested to tackle this problem. Some of them (19) exploit specific visual features that do not translate favorably to our problem. Others (20) employ tree search techniques resulting in exponential algorithm complexity. The most relevant techniques for our purpose are those known as alignment (21), pose-clustering (22), and geometric hashing (23) (for a comparison of these techniques, see ref. 24).

Recently, the geometric hashing paradigm for model-based object recognition was introduced by Lamdan *et al.* (23, 25, 26). This technique is especially geared toward recognition of partially occluded objects belonging to large-object data bases, and its complexity is a low-degree polynomial in the objects size. It is also very well suited for massive parallel implementation, and prototypes of this algorithm have been implemented on the highly parallel connection machine (27, 28). Techniques derived from computer vision have not been yet applied to molecular biology. We believe that their application will result in a significantly better performance than the manual graphics methods currently used not only because they introduce a fully automated approach but also because they have a key ability to detect patterns not known *a priori*.

The algorithm presented here includes automated scanning of a large number of structures. It assumes no *a priori* predefined motif. It is a true geometrical 3D comparison algorithm and thus is completely independent of the order of the amino acids in the primary chain. Furthermore, since the algorithm is sequence-independent, it is insensitive to gaps, insertions, or deletions, which constitute a major difficulty in structural comparisons based on sequence alignments. In principle, it can be implemented for both structure-related sequence motifs [sequence patterns that are associated with a specific structure (29)] and structural motifs (whose actual sequences may vary). It is general and can be used on both molecular model and crystal structure data. In addition to atomic coordinates, such a data base should preferentially also contain consistently defined sets of properties, such as secondary structures and hydrogen bonding (29). Several such data bases are being developed. Our algorithm can use protein or DNA/RNA structures, atomic labels, conformation coordinates, secondary structures, and tertiary interactions (29) in its structural comparisons (30). The more information included in the data base; the faster is the comparison.

Although growing fast, the B-DNA crystal structure library is still limited. Currently there are several DNA structural computation schemes (e.g., refs. 31–34). The RNA structural information is mostly derived from tRNA crystal structures.

A version of our proposed algorithm has been applied (35) to proteins that have been compared using other methods. It recovered all the alignments that have been obtained by the other methods, but whereas all these other methods used some additional information, which has been crucial for their success, our algorithm used no prior assumptions.

The Geometric Hashing Paradigm

The geometric hashing paradigm for model-based object recognition was introduced by Lamdan *et al.* (23, 25). Efficient algorithms were developed for recognition of rigid objects both in two and three dimensions.

We present here a variant of the geometric hashing technique for recognition of identical partial structures in rigid 3D objects. For the moment we will use purely geometric language whose biological equivalents are as follows. A (geometric) rigid object is analogous to a molecule. Such an object consists of a set of points, which correspond to atoms. Each point may have a label (the name of an atom). Given a data base of known objects (molecules) and an observed object, the algorithm finds those objects in the data base, having large substructures nearly identical with substructures of the observed object. The points of matched substructures should have equivalent labels and identical 3D coordinates modulo translation and rotation (rigid motion) in space. No *a priori* knowledge of the desired substructure is assumed.

In a model-based object recognition system, one has to address two major interrelated problems, namely, object representation and matching. The representation used must be rich enough to allow reliable distinction between the different objects in the data base, yet terse enough to enable efficient matching. A major factor in a reliable representation scheme is its ability to deal with recognition of partial substructures. In the geometric hashing technique objects are represented as sets of geometric features (in our case, points), and their geometric relations are encoded using minimal sets of such features under the allowed object transformations (in our case, rigid motion). This is achieved by standard methods of analytic geometry invoking coordinate frames based on a minimal number of features and representing other features by their coordinates in the appropriate frame. In the sequel we present the geometric hashing method for 3D point matching under translation and rotation.

The substructure recognition problem can be rephrased to the following point-set matching task, where one is given a set of known (model) point sets and an observed point set. The recognition task becomes the following subset isometry problem: Is there a rotated and translated subset of some model point set that matches a subset of the observed point set, so that both the geometric and labeling constraints are satisfied?

Representation of Geometric Constraints. Our goal is to represent a set of 3D points belonging to a rigid body by few intrinsic parameters. This representation should efficiently encode the geometric constraints of a rigid body, be translation and rotation invariant, and enable handling of partial (*a priori* unknown) substructure information.

Assume that we are given an arbitrary set of m points belonging to a rigid body. One can pick any ordered triplet of noncollinear points in the set and represent all the other points using this triplet. Specifically, let $\mathbf{e}_{00}$, $\mathbf{e}_{10}$, $\mathbf{e}_{01}$ be an ordered triplet of noncollinear points. These three points define a plane. One may choose an orthogonal 3D coordinate system centered at $\mathbf{e}_{00}$ such that the above mentioned plane is the (x, y) plane, the x axis is in the direction of the vector $\mathbf{e}_{10} - \mathbf{e}_{00}$, the y axis is orthogonal to it in the counterclockwise direction, and the z axis is orthogonal to the plane and its direction is defined by the right hand rule. Since we are dealing with rigid motion, the length of the unit vector can be predefined. Let $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$ be the relevant unit vectors. Any point $\mathbf{v}$ in the 3D space can be represented in the above mentioned coordinate system; namely, there is a triple of scalars (α, β, γ) such that $\mathbf{v} = \alpha\mathbf{e}_x + \beta\mathbf{e}_y + \gamma\mathbf{e}_z + \mathbf{e}_{00}$.

In the sequel we refer to the ordered triplet $(\mathbf{e}_{00}, \mathbf{e}_{10}, \mathbf{e}_{01})$ as a reference set (RS). Application of a rigid motion $\mathbf{T}$ will transform the point $\mathbf{v}$ to $\mathbf{Tv} = \alpha\mathbf{Te}_x + \beta\mathbf{Te}_y + \gamma\mathbf{Te}_z + \mathbf{Te}_{00}$.

It is easy to see that the triplet $(\mathbf{Te}_x, \mathbf{Te}_y, \mathbf{Te}_z)$ is an orthonormal 3D basis, which can be obtained from the RS $(\mathbf{Te}_{00}, \mathbf{Te}_{10}, \mathbf{Te}_{01})$ as above. Hence, the coordinates (α, β, γ) of a 3D point (atom) are invariant under a rigid motion.

Accordingly, we represent the m points of our model set by their coordinates in the basis triplet associated with the RS ($\mathbf{e}_{00}$, $\mathbf{e}_{10}$, $\mathbf{e}_{01}$).

The above mentioned representation allows comparison of partially identical objects. Assume that the observed object has a substructure that is identical to a substructure of one of the models in the data base. Then, the labels and the point coordinates of the observed object will have a partial overlap with the labels and corresponding coordinates of the stored model, if both are represented in a coordinate frame that is based on the same RS. Moreover, in our case one may eliminate many unsuitable RSs before the coordinates of the other points are computed. Two reference triplets have to be compared only if the triangles that they form are congruent and the corresponding vertices have matching labels.

The dependence of the representation on a specific RS may, however, preclude recognition when at least one of the RS points does not belong to the identical substructure. Hence we represent the object points by their coordinates in all basis triplets associated with all possible RSs. More specifically, given a model object, consisting of rigidly connected points, the labeling and geometric constraints describing the object are memorized in a hash table. The following preprocessing is applied to each model object.

For each ordered noncollinear triplet (RS) of model points (denoted by RS in the sequel) do the following operations: (*i*) Compute the orthonormal 3D basis associated with the RS. (*ii*) Compute the coordinates of all other m-3 model points in the coordinate frame defined by the 3D basis. (*iii*) For each such point define an address to a hash table with the labels and measurements (say three sides) defining uniquely the RS triangle and the label and coordinate (after a proper quantization) of the model point in the 3D basis. (*iv*) Use each such address (index) to enter the hash table and record in the appropriate entry the pair (model, RS), namely, the model and the RS for which this address was computed.

The complexity of this preprocessing step is of order m^4 per model. Each of the m points is represented in m^3 RSs. Although the dimensionality of the hash table (depending on the address) might seem high, one should remember that each new object can fill at most m^4 entries, so the actual space complexity of the hash table is $O(N \times m^4)$, where N is the number of the objects in the data base.

This somewhat redundant representation allows efficient matching of objects having only partial (previously unknown) equivalent substructures.

Note, that the preprocessing step is done without any knowledge of the observed object that has to be compared with the data base. Hence, it can be executed off-line, so that its execution time does not add to the actual recognition time. New models added to the data base can be processed independently without recomputing the hash table. The hash table preparation stage may be viewed as a learning stage of the algorithm. In this stage relevant information of the models is memorized in its various representations.

Matching. The matching stage of the algorithm uses the hash table, prepared in the representation (learning) stage. Given an observed object, one chooses a reference triplet, computes the coordinates of other points in the basis associated with this triplet, and tries to match their labels and coordinates to those memorized in the hash table. Specifically, one does the following operations:

(*i*) Choose an RS and compute the 3D basis associated with it. (One might, of course, try some "intelligent" choice of the RS, rather than choosing it at random. This might be appropriate if there is biological evidence for the existence of certain groups of atoms in typical motifs.)

(*ii*) Compute the coordinates of the other observed object points in the 3D basis.

(*iii*) For each such point enter the hash table at the address defined by the labels and measurements of the RS triangle and the label and coordinate of the new point. For every pair (model, RS) that appears in the entry of the hash table, tally a vote for the model and the RS as corresponding to the pair that was chosen on the observed object. (The accumulator of the votes will have $\Sigma_{i=1}^{N} m_i^3$ entries, where N is the number of models and m_i is the number of points on the ith model.)

(*iv*) If no pair (model, RS) in the hash table scores high, go back to step *i* and begin the procedure with a different RS of the observed object. If a certain pair (model, RS) scores a large number of votes (according to some predefined threshold), decide that this pair corresponds to the one chosen on the observed object. The uniquely defined rigid motion between the 3D coordinate frames, associated with the corresponding RSs, is the transformation between the appropriate model and the observed object.

(*v*) Consider all the model observed object point pairs that voted for the rigid motion (translation and rotation) of step *iv* and find the rigid motion giving the best least-squares match between all these corresponding point pairs. Since the computation of this transformation is based on more than three point pairs, it will be more reliable.

(*vi*) Transform the model point set according to the transformation of step *v*, align it with the new observed object, and check consistency of all the available biological information, such as relevant chemical links between atoms, etc. If this final verification fails, go back to step *i* and begin the procedure for a different observed object RS.

It is important to mention that in general we do not expect the voting scheme to give only one candidate solution (for an analysis in a more difficult computer vision application, see ref. 36). The goal is to reduce significantly the number of possible candidates for the verification step *vi*, which might be quite tedious and time consuming.

Since the voting is done simultaneously for all models and all possible RSs on a model, for the algorithm to be successful it is enough to pick three points on the observed object, belonging to some model. In such a case the model with the appropriate RS gets a high score in the voting procedure. The voting process, per RS, is linear in the number of points on the observed object. Hence, the overall recognition time is dependent on the "density" of model points in the observed object. Although, in the worst case, we might have an order of n^4 operations (assuming constant processing in each hash table bin), in most cases the recognition will be much faster, due to the very powerful RS congruence and atom label coincidence constraints. One may enhance the voting procedure by introducing a weighted voting scheme (see ref. 37). Namely, instead of giving an equal vote to each hash table bin, one may assign a high vote to bins with a small number of candidates, and a low vote to bins with a large number of candidates. In such a way rare configurations will get a higher vote than frequent ones. The weighted voting approach can also improve the efficiency of our algorithm. By assigning zero weight to bins with candidates above a certain threshold, we can save the time needed to process hash table entries with a lot of candidates. These entries require much computer time but contribute only a small amount of information. Hence, we may assume constant processing in each hash table bin.

The presented method can be parallelized in a straightforward manner. It has few serial steps, but most of the work can be done in parallel. Several versions of it for computer vision applications have been already implemented on the highly parallel connection machine (27, 28). It should also be quite easy to build special hardware for this purpose. As was mentioned before, the learning (hash table preparation) stage is independent of the actual recognition stage.

Improvements of the Basic Paradigm. In the previous section we have described the basic geometric hashing scheme for 3D substructure detection. Various improvements are possible. In particular, one can design an $O(n^3)$ worst case algorithm for that purpose, although the practical run time of the previous version should also be much less than its worst case estimate. This other version is also more space efficient and requires a hash table of $O(n^3)$ only. On the other hand, we use somewhat weaker geometric and labeling constraints. In this section we sketch this second more efficient (in the worst case) algorithm.

In the scheme described above, we used full 3D bases that were associated with three-point RSs. One may, however, use somewhat weaker information; namely, two-point RSs. Given a two-point RS, any other (noncollinear) point in the 3D space defines a plane with this RS. Compute the two-dimensional (2D) coordinates of this point in the above mentioned plane using a 2D orthonormal coordinate frame, which is associated with the RS (the first point is the origin, and the vector from it to the second point defines the x axis). The address to the hash table this time will be the labels and the length of the RS segment, and the label and the 2D coordinates (in the appropriate plane) of the point. Since this procedure is done for all reference pairs, the hash table will take $O(n^3)$ space.

The recognition stage will be similar to the previous version, only this time one has to pick a reference pair on the observed object instead of reference triplet. Hence the worst case complexity reduces to $O(n^3)$. Since weaker geometric and labeling constraints are applied in this version, one may expect somewhat more candidate solutions passing the first voting stage. This ambiguity will be easily resolved in the least squares and final verification steps of the algorithm.

Yet another way to reduce the computational load is to apply the algorithm (in the first stage) to C^α atoms only. Besides the reduction of computation, it also allows us to base the comparisons on stable structures. Such an approach, however, does not allow us to apply labeling constraints.

A significant improvement in the efficiency of the algorithm can be achieved by taking groups of atoms rather than single atoms as primitive building blocks of the substructures. In such a case, a single group may serve as a natural RS (if it has more than three noncollinear atoms), thus improving both the space and run time worst case complexity to $O(n^2)$, where n is the number of groups. For example, in the DNA natural primitive atom groups are adenine, cytosine, guanine, and thymine. Obviously, a generalized version of our algorithm can handle both atom groups and single atoms.

Experimental Results

A version of the proposed algorithm has been applied (35) to proteins that have previously been compared using other methods. In particular, we have implemented an improved version of the algorithm that compared only C^α atoms and used two-point RSs, as described in the previous subsection. Specifically, our technique has been used in the following experiments.

(*i*) To find nonpredefined similar domains in bacterial ferredoxin from *Peptococcus aerogenes*. Excellent fit of our results with those of Rossman and Argos (38) has been obtained.

(*ii*) Two members from the phospholipase A_2 proteins were compared—phospholipase A_2 from bovine pancreas and *Crotalus atrox* venom. These proteins have been previously compared by Renetseder *et al.* (39) using standard techniques (i.e., finding "by eye" a similar core and then aligning using the least-square procedure). Again, our alignment corresponds exactly to that reported by Renetseder *et al.* (39).

(*iii*) The HTH motif was located in several bacterial repressor proteins just as noted in the annotated protein data bank (PDB). In our experiments we have compared three transcriptional regulatory proteins known to contain the HTH motif: tryptophan repressor (PDB code, 2WRP), λ Cro (PDB code, 1CRO), and phage 434 Cro (PDB code, 2CRO). To give a flavor of our experimental results we describe this example in more detail.

In 1CRO, there are four crystallographically unrelated monomers in the asymmetric unit. These monomers have been assigned chain identifiers O, A, B, and C. The dimer of 1CRO that exists in solution is presumed to be the O–B dimer, which is thought to be the one that actually binds DNA. We use the B monomer in the comparisons shown below, but comparisons using all four domains produce similar matches.

The sequence positions where the HTH motifs appear are as follows:

Protein	Positions	Sequence
2WRP	66–88	MS QRELKNELGA GIATITRGSNS
1CRO	14–36	FG QTKTAKDLGV YQSAINKAIHA
2CRO	15–37	MT QTELATKAGV KQQSIQLIEAG

In the three pairwise comparisons below (see Table 1), our method succeeds in matching the HTH motif from one protein to the HTH motif from the other. Very few other atom pairs are matched, showing that the only equivalent substructure between the proteins is the HTH motif itself. The atom pairs outside the HTH motif are 3D nonlinear matches.

For each pair of matching substructures Table 1 gives the sequence numbers of the matching atoms, the transformation between the substructures (translation parameters in angstroms and rotation angles in radians), and the rms distance between the matching substructures subject to the appropriate transformation. In these examples one of the proteins was taken as the data base (model) and the other protein was as the unknown structure (scene).

Although the HTH motif does conserve the linear sequence structure, our algorithm did not exploit this assumption but rather tackled the problem as a 3D-matching problem. Moreover, it had no *a priori* information that it was the HTH motif we were looking for.

(*iv*) Two proteins from the calmodium/calcium binding protein group were compared—parvalbumin and intestinal calcium binding protein. Several matches were obtained. Two of these correspond to the alignment reported by Taylor and Orengo (40).

(*v*) Bovine liver rhodanese contains two motifs, which have been compared both by Taylor and Orengo (40) and by Ploegman *et al.* (41), yielding similar results. Our matches are almost identical to those obtained (40, 41).

(*vi*) Two lysozymes have been compared from hen egg whites and T4 phage. Our matches compare favorably with those of Rossman and Argos (38), Weaver *et al.* (42), and Taylor and Orengo (40).

Details of the programming, results, and their comparisons with the previously published results are presented elsewhere (35). It should, be noted, however, that previously published matches are based on linear sequence structural comparisons, where contiguous amino acids are matched. Our 3D comparisons had no such prior assumptions and have also unraveled some real 3D sequence-order-independent matches. We expect that intensive applications of the method to the crystallographic data base will yield additional recurring spatial motifs.

Conclusion and Future Research

We have presented an algorithm for structural comparisons. As the computational approaches and structural predictions of DNA, RNA, and in particular, proteins improve (43), such

Table 1. Pairwise matchings of the HTH motif in three proteins: 2CRO (phage 434), 1CRO (λ phage), and 2WRP (tryptophan repressor)

Model 2CRO	Scene 2WRP	Model 2WRP	Scene 1CROB	Model 1CROB	Scene 2CRO
				55-V	60-Q
					—
	—			44-I	53-N
63-T	103-V				—
	—			51-Y	50-M
				52-A	49-A
	—		—		—
37-G	88-S	88-S	36-A	36-A	37-G
36-G	87-N	87-N	35-H	35-H	36-A
35-E	86-S	86-S	34-I	34-I	35-E
34-I	85-G	85-G	33-A	33-A	34-I
33-L	84-R	84-R	32-K	32-K	33-L
32-Q	83-T	83-T	31-N	31-N	32-Q
31-I	82-I	82-I	30-I	30-I	31-I
30-S	81-T	81-T	29-A	29-A	30-S
29-Q	80-A		28-S	28-S	29-Q
28-Q	79-I	79-I	27-Q	27-Q	28-Q
27-K	78-G	78-G	26-Y	26-Y	27-K
26-V	77-A	77-A	25-V	25-V	26-V
25-G	76-G	76-G	24-G	24-G	25-G
24-A	75-L	75-L	23-L	23-L	24-A
23-K	74-E	74-E	22-D	22-D	23-K
22-T	73-N	73-N	21-K	21-K	22-T
21-A	72-K	72-K	20-A	20-A	21-A
20-L	71-L	71-L	19-T	19-T	20-L
19-E	70-E	70-E	18-K	18-K	19-E
18-T	69-R	69-R	17-T	17-T	18-T
17-Q	68-Q	68-Q	16-Q	16-Q	17-Q
16-T	67-S	67-S	15-G	15-G	16-T
	66-M	66-M	14-F	14-F	15-M
13-L	65-E	65-E	13-R	13-R	14-K
	64-G	64-G	12-M		13-L
9-R	63-R		11-A		12-A
	62-L	60-E	10-Y		11-I
	61-L	63-R	9-D	10-Y	10-R
	60-E	61-L	8-K		9-R
	59-E		7-L		8-K
	58-V	59-E	6-T	8-K	7-K
	57-I			7-L	6-L
	—		—		—
44-F	53-T			39-K	2-L
	—				—
43-R	50-A				

Columns 1 and 2: translation, 16.6, −7.7, −2.4; rotation, −0.11, 0.29, −2.53; rms, 0.90. Columns 3 and 4: translation, −16.6, −49.6, −20.1; rotation, −2.34, 0.64, −0.90; rms, 1.29. Columns 5 and 6: translation, −23.4, −45.2, −19.7; rotation, 1.91, −0.77, −2.59; rms, 0.97.

an algorithmic tool, borrowed and adapted from computer vision, can be very extensively implemented. We have implemented a preliminary version of the geometric hashing for molecular biology applications on a serial computer. The initial experiments show considerable promise.

We view the presented algorithms only as a basic paradigm. Additional biological information can be incorporated into this basic framework. In particular, one can consider the chemical links between various atoms and groups of atoms. Any such additional information adds additional matching constraints and may speed up the algorithm.

1. Nussinov, R. (1990) *CRC Crit. Rev. Biochem. Mol. Biol.* **25**, 185–224.
2. Pabo, C. O. & Sauer, R. T. (1984) *Annu. Rev. Biochem.* **53**, 293–321.
3. Klug, A. & Rhodes, D. (1987) *Trends Biochem. Sci.* **12**, 464–469.
4. Gehring, W. J. (1987) *Science* **236**, 1245–1252.
5. Landschulz, W. H., Johnson, P. F. & McKnight, S. L. (1988) *Science* **240**, 1759–1764.
6. Murre, C., McCaw, P. S. & Baltimore, D. (1989) *Cell* **56**, 777–783.
7. Suzuki, M. (1989) *EMBO J.* **8**, 797–804.
8. Mermod, N., O'Neill, E. A., Kelly, T. J. & Tjian, R. (1989) *Cell* **58**, 741–753.
9. Courey, A. J. & Tjian, R. (1988) *Cell* **55**, 887–898.
10. Tanaka, I., Appelt, K., Dijk, J., White, S. W. & Wilson, K. S. (1984) *Nature (London)* **310**, 376–381.
11. Rafferty, J. B., Somers, W. S., Saint-Girons, I. & Phillips, S. E. V. (1989) *Nature (London)* **341**, 705–710.
12. Abel, T. & Maniatis, T. (1989) *Nature (London)* **341**, 24–25.
13. Sankoff, D. & Kruskal, J. B. (1983) *Time Warps, String Edits and Macromolecules* (Addison-Wesley, Reading, MA).
14. Mitchel, E. M., Artymiuk, P. J., Rice, D. W. & Willet, P. (1989) *J. Mol. Biol.* **212**, 151–166.
15. Richards, F. M. & Kundrot, C. E. (1988) *Protein Struct.* **3**, 71–84.
16. Abagyan, R. A. & Maiorov, N. V. (1988) *J. Biomol. Struct. Dyn.* **5**, 1267–1279.
17. Besl, P. J. & Jain, R. C. (1985) *ACM Comput. Surv.* **17**, 75–154.
18. Chin, R. T. & Dyer, C. R. (1986) *ACM Comput. Surv.* **18**, 67–108.
19. Bolles, R. C. & Horaud, P. (1986) *Int. Robotics Res.* **5**, 3–26.
20. Grimson, W. E. & Lozano-Pérez, T. (1987) *IEEE Trans. Pattern Anal. Machine Intelligence* **9**, 469–482.
21. Huttenlocher, D. P. & Ullman, S. (1988) *Proceedings of the DARPA Image Understanding Workshop* (Morgan Kaufmann, San Mateo, CA), pp. 1114–1122.
22. Stockman, G. (1987) *J. Comput. Vision. Graphics. Image Process.* **40**, 361–387.
23. Lamdan, Y., Schwartz, J. T. & Wolfson, H. J. (1988) *Proceedings of IEEE International Conference on Robotics and Automation* (IEEE, New York), pp. 1407–1413.
24. Wolfson, H. J. (1990) in *Proceedings of the European Conference on Computer Vision*, ed. Faugeras, O. (Springer, Berlin), pp. 526–536.
25. Lamdan, Y. & Wolfson, H. J. (1988) *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, New York), pp. 238–249.
26. Lamdan, Y., Schwartz, J. T. & Wolfson, H. J. (1990) *IEEE Trans. Robotics Automation* **6**, 578–589.
27. Bourdon, O. & Medioni, G. (1990) *Proceedings of International Conference on Pattern Recognition* (IEEE, New York), pp. 596–600.
28. Rigoutsos, I. & Hummel, R. A. (1991) *IEEE Workshop on Directions in Automated CAD-Based Vision* (IEEE, New York), pp. 76–84.
29. Thornton, J. M. & Gardner, S. P. (1989) *Trends Biochem. Sci.* **14**, 300–304.
30. Sutcliffe, M. J., Hanaeef, I., Carney, D. & Blundell, T. L. (1987) *Protein Eng.* **1**, 377–384.
31. Sarai, A., Mazur, R., Nussinov, R. & Jernigan, R. L. (1988) *Biochemistry* **27**, 8498–8502.
32. Strinivasan, A. R., Torres, W., Clark, R. & Olson, W. K. (1987) *J. Biomol. Struct. Dyn.* **5**, 459–496.
33. Tung, C. S. & Harvey, S. C. (1986) *J. Biol. Chem.* **261**, 3700–3709.
34. Ulyanov, N. B. & Zhurkin, V. B. (1984) *J. Biomol. Struct. Dyn.* **2**, 361–385.
35. Fischer, D., Bachar, O., Nussinov, R. & Wolfson, H. J. (1991) *J. Biomol. Struct. Dyn.*, in press.
36. Lamdan, Y. & Wolfson, H. J. (1991) *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference* (IEEE, New York), pp. 22–27.
37. Hong, J. & Wolfson, H. J. (1988) *Proceedings of International Conference on Pattern Recognition* (IEEE, New York), pp. 72–78.
38. Rossman, M. G. & Argos, P. (1976) *J. Mol. Biol.* **105**, 75–96.
39. Renetseder, R., Brunie, S., Dijkstra, B. W., Drent, J. & Sigler, P. B. (1985) *J. Biol. Chem.* **206**, 11627–11634.
40. Taylor, W. R. & Orengo, C. A. (1989) *J. Mol. Biol.* **208**, 1–22.
41. Ploegman, J. H., Drent, G., Kalk, K. H. & Jol, W. G. (1987) *J. Mol. Biol.* **123**, 557–594.
42. Weaver, L. H., Grutter, M. G., Remington, S. J., Gray, T. M., Issacs, N. W. & Matthews, B. W. (1985) *J. Mol. Evol.* **21**, 97–111.
43. Jernigan, R. L., Sarai, A., Covell, D. G. & Mazur, J. (1988) in *Journal of International Conference on Supercomputers* (Boston), Vol. 1, pp. 197–200.