BIOINFORMATICS

Vol. 14 no. 6 1998
Pages 516–522

A geometric algorithm to find small but highly similar 3D substructures in proteins

Xavier Pennec[1] and Nicholas Ayache

INRIA, BP 93, 2004 route des Lucioles, 06902 Sophia Antipolis Cedex, France

Received on December 9, 1997; revised and accepted on February 24, 1998

Abstract

Motivation: *Most biological actions of proteins depend on some typical parts of their three-dimensional structure, called 3D motifs. It is desirable to find automatically common geometric substructures between proteins to discover similarities in new structures or to model precisely a particular motif. Most algorithms for structural comparison of proteins deal with large (fold) similarities. Here, we focus on small but precise similarities.*

Results: *We propose a new 3D substructure matching algorithm based on geometric hashing techniques. The key feature of the method is the introduction of a 3D reference frame attached to each residue. This allows us to reduce drastically the complexity of the recognition. Our experimental results confirm the validity of the approach and allow us to find smaller similarities than previous methods.*

Availability: *The program uses commercial libraries and thus cannot be completely freely distributed. It can be found at ftp://www.inria.fr in the directory epidaure/Outgoing/xpennec/Prospect, but it requires a key to be run, available by request to xavier.pennec@sophia.inria.fr*

Contact: *Xavier.Pennec@sophia.inria.fr; Nicholas.Ayache@sophia.inria.fr*

Introduction

Most biological actions of proteins, such as catalysis or regulation of the genetic message (transcription, maturation, etc.), depend on some typical parts of their three-dimensional structure, called 3D structural or binding motifs. Proteins with similar 3D motifs often show similar biological properties, and it is therefore highly desirable to find similar 3D motifs between proteins (Branden and Tooze, 1991). Since proteins are composed of possibly thousands of atoms, the search requires efficient and fully automated methods.

There is quite an extensive literature on 3D protein structure comparison. Early techniques, such as Rossmann and Argos (1976) and Remington and Matthews (1980), required seed matches and tried to align the entire structures in 3D. Then, a series of algorithms focused on backbone fragment similarities, first finding compatible fragments and then extending and clustering them in more global matches. Examples of such algorithms can be found in Alexandrov *et al.* (1992) and Lessel and Schomburg (1994). Holm and Sander (1994) give a good review of available techniques at that time. A more recent trend, reflected in Holm and Sander (1995), Madej *et al.* (1995) and Alexandrov and Fischer (1996), reviewed in Gibrat *et al.* (1996), is to model a protein by the set of its secondary structure elements (SSEs) and identify very rapidly the matches between SSEs using binary geometric constraints. Then, an exhaustive search for compatible SSE matches is performed using interpretation trees, maximal clique or clustering algorithms [see Grimson (1990) for a review of geometric matching algorithms].

However, while these techniques are well adapted to detecting large structural similarities (folds or topological similarities), it has been argued (Mizuguchi and Go, 1995; Gibrat *et al.*, 1996) that similarities of small proteins with few or no secondary structure elements may not be detected at all: even a precisely conserved motif will go essentially unnoticed if it does not include enough α helices or β strands. To look for such similarities, we have to focus on the 3D configuration of residues in space and forget their primary and secondary structures.

In this spirit, Fischer *et al.* (1992) and Bachar *et al.* (1993) have exploited the geometric hashing paradigm previously introduced in computer vision by Lamdan and Wolfson (1988) and Wolfson (1990). They proposed substructure matching methods based on pre-processing and recognition algorithms of complexity $O(n^3)$, where n is the number of residues of interest (either in the motif or in the protein). A key point of their approach is the possibility to refer to two rigid invariants (the 'distance coordinates') of any residue of the protein with respect to two other residues picked arbitrarily as forming a geometric 'basis'. The results reported in their publications were encouraging, and motivated our work.

Our main idea, introduced in Pennec and Ayache (1994b), was to reduce the size of a 'basis' from two to a single residue. To achieve this goal, we introduce a 3D reference frame

[1]*To whom correspondence should be addressed*

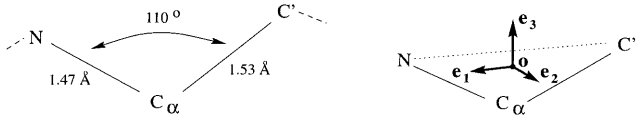

Fig. 1. Geometry of a residue around the C_α and definition of a basis.

attached to each one. Doing this, we can now choose a single residue as a basis, and compute six rigid invariants (the parameters of translation and rotation) attached to any other residue. This allows us to reduce drastically the complexity of both the pre-processing and recognition stages of geometric hashing, typically from $O(n^3)$ to $O(n^2)$. The idea of using frames instead of just the C_α position to represent amino acids was also proposed in Boutonnet *et al.* (1995) and Orengo and Taylor (1996).

A thorough analysis of the propagation of the uncertainties in the computation of invariants, transformation estimation, and clustering (Pennec, 1996; Pennec and Thirion, 1997) guided our implementation to ensure efficiency and robustness of the approach. Our experimental results confirm the validity of the approach, and show that we can detect smaller similarities than previous methods.

The paper is organized as follows: first we detail the reference frame attached to each residue, and then we describe the new geometric hashing algorithm we propose for matching. Third, we report our experimental study. Finally, we present some potential extensions for our work.

Protein structure modeling

Topologically, the backbone of the chain is linear, but its geometry is more complex. Rotations are allowed around the bonds C_α–C and C_α–N, and hence the geometry of the chain is weakly constrained. However, the geometry of the atoms attached to the C_α is perfectly determined. In particular, the three atoms N, C_α, C form a known triangle from which we can define a frame (a point and a trihedron; see Figure 1), which uniquely defines the position and orientation of the residue in space. We will hence model a residue by a couple (point, trihedron) and a protein by the set of these frames.

The structure comparison problem is thus stated as follows: given two sets of frames, find all rigid transformations that match a minimum number of residues of the two structures. We delay the problem of the classification criterion and the assessment of the matches' significance until the Discussion. The problem can be extended to the comparison of a target molecule with a database of proteins.

Matching proteins

The problem we are confronted with is very close to recognition problems in volume image analysis, especially in the medical field. In this case, one has to process points extracted from surfaces with their associated Frénet trihedron (Thirion, 1996; Guéziec *et al.*, 1997). In both cases, the model adopted to reduce the data is a set of frames. Classical techniques rely on a model-based approach for object recognition (Grimson, 1990). Given a database of modeled objects (called models), the aim is to recognize in a scene what objects are present, and how they are placed. The simplest problem where the database is reduced to only one object is called simply matching or sometimes registration.

The geometric hashing algorithm

The geometric hashing algorithm was introduced (Lamdan and Wolfson, 1988; Wolfson, 1990) for model-based recognition in computer vision. The basic idea is to store in a database at pre-processing time a redundant representation of models, based on local features to allow for occlusion and invariant by rigid transformation. By doing so, the representation of the scene computed at recognition time will present some similarities with that of some database objects. Accumulating this evidence will allow the recognition and registration of objects present in the scene and in the database.

Invariant description. In our case, local features are frames. However, any model frame can be matched with any scene frame. Thus, to obtain an invariant description, we have to consider binary constraints between frames. Indeed, a pair of frames has six invariants given by the rigid transformation parameters from the one frame to the other (expressed in one of the frames).

In order to deal with occlusion, the representation of one frame has to be redundant: each frame will then be associated with any other frame of the object to compute the set of 6D invariant vectors characterizing this reference frame. The global representation is then the set of every frame pair of the model, each one being an entry for the hash table, with the 6D invariant vector as index.

Pre-processing. In order to optimize the access to the representation for recognition time, the geometric hashing algorithm uses a hash table for storing models. Indeed, given one object, we just compute the 6D invariant vector associated with each possible pair (the reference frame, another model frame), and set it as an index in a 6D hash table for the pair. Each model is processed independently, but stored in the same hash table. The complexity of the step is $O(Mm^2)$, where M is the number of models and m the mean number of residues per model. Typical values for m range from <15 for template motifs to a few hundreds for big proteins. The complexity in space for the hash table is the same since it only depends on the number of entries. This step is performed without any knowledge of the scene to be matched and hence can be done once for all.

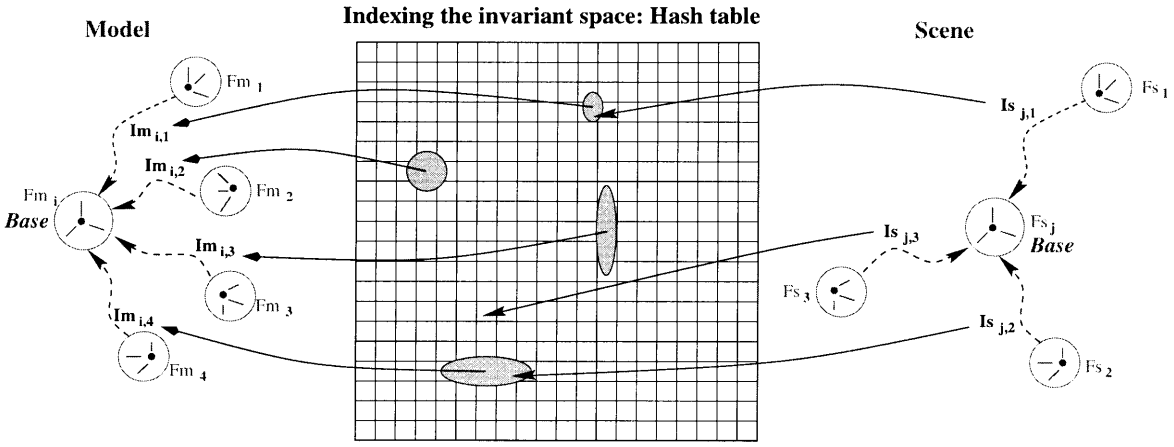

Fig. 2. Pre-processing: the 6D invariant vector associated with every model frames pair is computed with its error zone and used as an index for the pair in the hash table. Recognition: for each scene frame pair, we compute the 6D invariant vector and retrieve through the hash table every compatible model frame pair. For each such pair, we tally a vote for the matching of the reference frames [here the match (Fm_i, Fs_j) scores 2].

Recognition. Choose a reference frame; for each different scene frame, compute the 6D invariant vector and retrieve the compatible model pairs (the reference frame, another model frame) in quasi constant time thanks to the hash table. During the process, maintain a list of the model reference frames found, and for each one accumulate the number of compatible pairs. This will be the score for the matching of these model reference frames with the considered scene reference frame.

The process is repeated with each scene frame taken as the reference frame. The output is the list of model and scene matching reference frames with their associated score (see Figure 2). We only keep the matches with a score above a threshold. This parameter is either static or dynamically adapted during the algorithm. It is also possible to keep a fixed number of matches (usually the best ones).

Error handling. Because of the resolution of the determination of protein structure, conformational deformations, and even structural differences between molecules that induce different constraints on the motif, one has to deal with errors in atom positions. Hence, each residue frame is given an associated covariance matrix, which is propagated through the computations (Pennec, 1996).

Since we now have probabilistic invariants, we should index and retrieve them using their error zone, which is defined as the uncertainty ellipsoid at a given χ^2. A statistical study shows that when the bin size of the hash table is more or less the error zone size, we have a mean number of 2^d bins intersecting the error zone, where d is the number of invariants (six in our case). From a computational point of view, we have found that replacing such an 'uncertain hash table' by KD trees (Preparata and Shamos, 1985) was more efficient for retrieving compatible invariants.

Clustering

The correspondence between a model reference frame and a scene reference frame is sufficient to compute a rigid transformation between the two proteins, but it is not very precise. During the recognition step, every compatible pair brings in some additional information (the matching of secondary frames). We use this information to refine the transformation between the model and the scene basis at a small cost using an extended Kalman filter (Pennec and Thirion, 1997).

Actually, the matches belonging to a common substructure will present a similar transformation. We can then regroup them by clustering their transformations [this idea was previously used in Vriend and Sander (1991)]. We first classify matches by decreasing information (the information of a random transformation is the opposite of the log of the determinant of its covariance matrix). Then, we choose the most informative transformation among the set to cluster and iteratively merge the closest compatible transformation to the current state estimate [according to the Mahalanobis distance $\mu^2(x,y) = (x-y)^T (\Sigma_x + \Sigma_y)^{-1} (x-y)$ and the χ^2 test; Pennec and Thirion, 1997]. Each used transformation is removed from the set. Once there are no more transformations to merge, we have obtained one cluster represented by its mean transformation and we iterate the clustering stage on the remaining transformations. In this process, an efficient way of finding the nearest (Mahalanobis) neighbor would be an important improvement for the complexity. A more rigorous algorithm would be to let the different clusters compete with

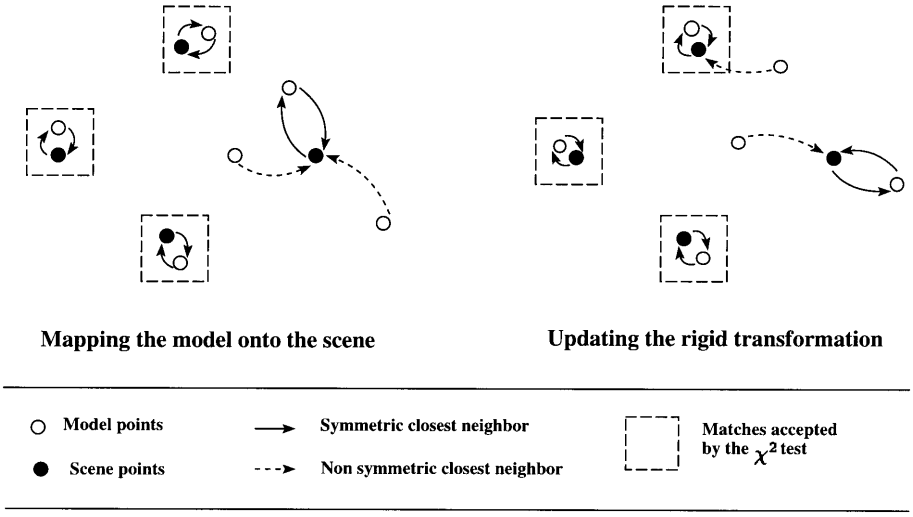

Fig. 3. Left: the model is mapped onto the scene and three matches are satisfying the symmetric closest point and χ^2 test constraints. Right: these three matches are used to recompute the rigid transformation and update the position of the model. Since the same matches are s atisfying the constraints, the algorithm can stop.

each other and to compute the mean transformation of a cluster with a Mahalanobis distance minimization at each step. As for the geometric hashing step, we only keep clusters that have a minimal number of matches. The complexity is $O(k_m k_c)$ with a number k_m of matches and k_c of clusters on output, but k_c is quasi constant in practice, a few dozens at the very most.

Verification and extension

Clusters must now be checked and their matching list extended. This is done using an alignment test: using the rigid transformation previously determined, the model is mapped onto the scene and the possible matches are verified. Each frame of the model is examined as follows (Figure 3).

Map the model frame onto the scene and search for the closest frame of the scene. In order to keep the algorithm symmetrical between the model and the scene, map back the scene frame to the model and verify that the original model frame is its closest neighbor. If not, reject the model frame.

Compute the Mahalanobis distance between the transformed model frame and the scene one, and decide using a χ^2 test whether this match is valid. If not, reject the model frame.

Update the rigid transformation of the cluster with this new match using the extended Kalman filter.

This process is repeated until convergence (stability of matches) or a maximum number of iterations (we experimented using 10). Seeking the closest neighbor is performed using k-D trees (Preparata and Shamos, 1985). The complexity of constructing a k-D tree is $O(n \log n)$ with $O(n)$ storage. The search for a closest neighbor is sub-linear, and almost constant in practice. Hence, the whole stage has a complexity $O(n \log n + nk)$.

Algorithm analysis

Simplifications. The time- and memory-consuming step in this algorithm is the creation, pre-processing and retrieval of binary invariants. We have m^2 such invariants for the model and n^2 for the scene. With an ideal hash table, the pre-processing step would be $O(m^2)$ and the recognition step $O(n^2)$ thanks to a constant access time to compatible invariants through the hash table. With the introduction of noise in measurements, the actual complexity is much higher. Practically, we have found that using KD trees was more efficient in time and memory: the pre-processing stage is now in $O(m^2 \log m)$ and the recognition in $O(n^2 \log m)$.

However, the residues of a motif are usually close in space, and we can focus on pairs of frames with inter-distances under a threshold (typically around 20 Å). Moreover, we do not want to find matches within a single secondary structure (α helix or β strand). Thus, we do not index nor try to retrieve binary invariants within such structures. These two heuristics theoretically limit the complexity well below the above values.

Parameters. There are a small number of parameters that need to be adjusted in the algorithm. The major ones are the standard deviations σ_{pos} and σ_{rot} for the noise on residues position and orientation. To compare these values easily with the RMS and the mean angle after matching, we use the 3D values of the standard deviations, which means that the cova-

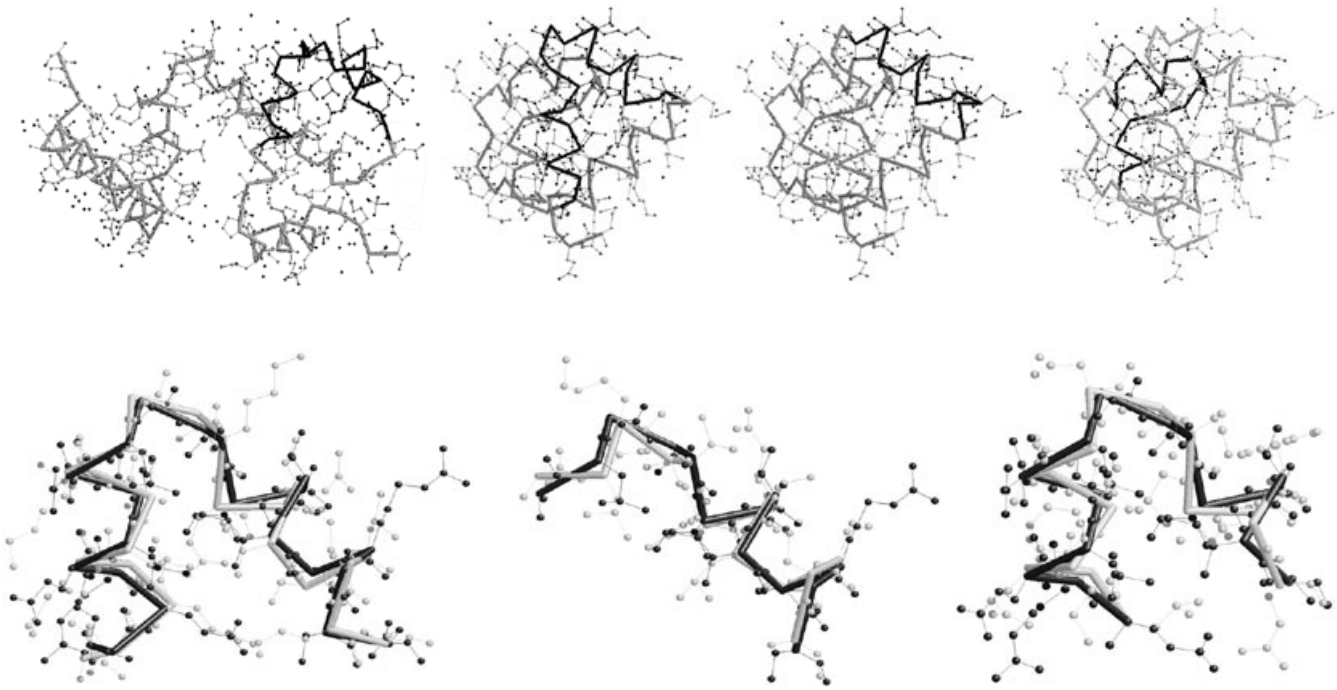

Fig. 4. Top, from left to right: the location of the HTH motif detected in 2WRP (tryptophan repressor of *E.coli*) and the locations of the substructures detected in 2CRO (CRO protein of phage 434). Bottom: registration of the three CRO substructures in the 2WRP coor dinate frame.

riance matrix of the rotation vector is $\Sigma_{\text{rot}} = \text{DIAG}(\sigma^2_{\text{rot}}/3, \sigma^2_{\text{rot}}/3, \sigma^2_{\text{rot}}/3)$ and similarly for the covariance matrix of the position. In our experiments, we usually use $\sigma_{\text{pos}} = 0.6$ Å and $\sigma_{\text{rot}} = 15°$. The related χ^2 test thresholds are χ^2_i for the binary invariant compatibility, χ^2_c for clustering transformations and χ^2_v on features for the verification step. Since all these values are relative to the same dimension (the dimension of rigid transformations in 3D is six), we use a single value $\chi^2 = \chi^2_i = \chi^2_c = \chi^2_v = 16$.

From the complexity point of view, the main parameter is the threshold $d_{\max}$ on the distance between residues to select binary invariants. We usually use $d_{\max} = 18$ Å. The last parameters are the minimal number of matches for the recognition step (n_r) and for clustering (n_c). We use $n_r = n_c = 5$. These values should be decreased to four or three to find smaller similar substructures.

Experiments

Comparison of 2CRO and 2WRP

We choose to compare the tryptophan repressor of *Escherichia coli* (PDB code 2WRP; Lawson *et al.*, 1988) and the CRO protein of phage 434 (PDB code 2CRO; Mondragon *et al.*, 1989), which are known to share a common substructure: the helix–turn–helix motif (Brennan and Matthews, 1989; Harrison and Aggarwal, 1990). This motif is responsible for the binding of DNA within several prokaryotic proteins. The atom coordinates of proteins are provided by Brookhaven National Laboratory's Protein Data Bank (Bernstein *et al.*, 1977; Abola *et al.*, 1987).

The execution time for the comparison of 2CRO and 2WRP is 18 s on a PC (pentium pro 200 MHz) running Linux. We have synthesized in Table 1 the output of the algorithm. The HTH motif is the most evident common substructure and is perfectly detected without spurious matches (37 GLY – 88 SER is not detected, but this match is indeed arguable considering the distance after registration and especially the difference in orientation). The two other detected substructures turn out to be quite interesting as they also match part of the HTH motif and were not previously detected. The second substructure is a kind of 'turn–helix' structure with the turn preceding the proper HTH motif of 2CRO. The last substructure is located farther in 2CRO and seems to be another HTH motif with shorter helices. We show in Figure 4 the location of the detected substructures in the two proteins and the registration between them. The images are made using the Rasmol program of R.Sayle (Sayle and Bissel, 1992) (the scripts for these visualizations are automatically produced by our program).

Table 1. Detected matches between the proteins 2CRO and 2WRP: synthesized output of the algorithm. Since the detected substructures are all part of the HTH motif in 2WRP, the matches are presented as a multiple alignment. Conserved residues are displayed in bold. The score is the information of the transformation. It is related to the mean Mahalanobis distance between matches divided by the number of matches

	2WRP	2CRO (1)	2CRO (2)	2CRO (3)
	66 **MET**	15 **MET**		
	67 SER	16 THR		
H	68 **GLN**	17 **GLN**		45 LEU
E	69 ARG	18 THR		46 PHE
L	70 **GLU**	19 **GLU**		47 **GLU**
I	71 **LEU**	20 **LEU**		48 **ILE**
X	72 LYS	21 **ALA**		49 **ALA**
	73 ASN	22 THR		50 MET
	74 GLU	23 LYS	12 **ALA**	51 **ALA**
T	75 **LEU**	24 ALA	13 **LEU**	52 **LEU**
U	76 **GLY**	25 **GLY**	14 LYS	53 ASN
R	77 ALA	26 VAL	15 MET	54 CYS
N	78 GLY	27 LYS	16 THR	55 ASP
	79 ILE	28 **GLN**	17 **GLN**	56 PRO
	80 ALA	29 GLN	18 THR	57 VAL
H	81 THR	30 SER	19 GLU	58 TRP
E	82 **ILE**	31 **ILE**	20 **LEU**	59 **LEU**
L	83 THR	32 **GLN**	21 ALA	60 **GLN**
I	84 ARG	33 LEU	22 THR	
X	85 GLY	34 ILE	23 LYS	
	86 SER	35 GLU	24 ALA	
	87 ASN	36 ALA		
Score		19.4	16.9	7.3
No. matches		22	13	16
RMS		0.66	0.65	0.81
Mean angle		15.9	16.1	22.8

Discussion

In order to test the sensitivity of the algorithm to the input parameters, we have also carried out the same experiment with a smaller expected noise on residues (σ_{pos} = 0.4 Å and σ_{rot} = 10°). The HTH motif was the only common substructure detected and only two matches are missing (28 GLN – R 79 ILE and 29 GLN – R 80 ALA).

With a larger expected noise (σ_{pos} = 0.8 Å and σ_{rot} = 20°), the HTH motif is still the first ranking substructure with an additional match (8 LYS – R 62 LEU). The second substructure detected is the 'small HTH' without modification. The 'turn–helix' structure now ranks 6, but is transformed into another small HTH with a new (but badly matched) α helix before the turn. In fact, although it presents 20 matches (from 6 LEU – R 68 GLN to 24 ALA – R 86 SER plus 53 ASN – R 59 GL), the RMS and the mean angle have been multiplied by two, which explains the lower score and ranking. The six other detected substructures are made of two α helices.

This shows that the use of frames instead of points improves the robustness of motif detection. Indeed, the orientation of a residue is crucial in determining the position of collateral chains and most protein interactions happen within these side atoms. The position of these atoms is then not only determined by the position of the backbone, but also by its orientation. Thus, using just points to represent residues generally leads to a significant increase in matches with non-compatible orientations, and implies a drastic reduction of selectivity for the matching process.

Other experiments show that the algorithm performs very similarly for detecting motifs based on β strands, for instance Greek key motifs or the β sheet DNA binding motif common to the arc repressor of *Salmonella* bacteriophage P22 (1ARR) and the Met repressor of *E.coli* (1CMA) (Raumann *et al.*, 1994).

From these experiments, we can see that our similarity criterion (the information, or 'accuracy', of the transformation) is only capturing the geometric properties and does not include a statistical analysis of the similarity significance. For instance, when comparing several structures, a few α helix or β strand matches are usually not significant, but only reflect the fact that two structures have such secondary structure elements. In order to scale up the method and compare a structure to the whole Protein Data Bank, we will need to incorporate a false-positive analysis, as in Grimson and Huttenlocher (1990), computing explicitly the probability of obtaining such a similarity score with random structures.

Conclusion

Modeling residues by the three atoms of their backbone allows us to define a complete and unique associated reference frame, which turns out to be very stable. Each residue pair hence has six invariants for rigid transformations that we use in a geometric hashing scheme to discover initial matches. These are clustered, verified and extended. The error inherent to the problem is integrated in the process, thanks to a rigorous theoretical framework for uncertainty handling. Experiments confirm the validity, efficiency and robustness of our approach.

This algorithm is also currently used for substructure matching in volume images (medical images) with frames extracted from surfaces (extremal points) (Guéziec *et al.*, 1997). This stresses the analogy between 3D matching problems and points out the fact that frames can, in numerous cases, advantageously replace points.

Future work will follow three axes. We plan first to use a probabilistic scheme for geometric hashing, for instance Rigoutsos and Hummel (1993), and incorporate a substitution matrix for matching residue types. A second improvement

would be a 'multi-scale analysis' of detected motifs, for instance studying size versus RMS, and a false-positive analysis to reject statistically non-significant similarities. Last but not least, the extension of our algorithm for multiple alignments, along with this more selective classification of similarities, could allow scanning of the Protein Data Bank or one of its representative sets.

References

Abola,E.E., Bernstein,F.C., Bryant,S.H., Koetzle,T.F. and Weng,J. (1987) Protein Data Bank. In Allen,F.H., Bergerhoff,G. and Sievers,R. (eds), *Crystallographic Databases—Information Contents, Software Systems, Scientific Applications.* Data Commission of the International Union of Crystallography, Bonn, pp. 107–132.

Alexandrov,N.N. and Fischer,D. (1996) Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins: Struct. Funct. Genet.*, **25**, 354–365.

Alexandrov,N.N., Takahashi,K. and Go,N. (1992) Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. *J. Mol. Biol.*, 5–9.

Bachar,O., Fischer,D., Nussinov,R. and Wolfson,H. (1993) A computer vision based technique for sequence independent structural comparison of proteins. *Protein Eng.*, **6**, 279–288.

Bernstein,F.C., Koetzle,T.F., Williams,G.J.B., Meyer,E.F., Brice,M.D., Rodgers,J.R., Kennard,O., Shimanouchi,T. and Tasumi,M. (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. *J. Mol. Biol.*, **112**, 535–542.

Boutonnet,N.S., Rooman,M.J., Ochagavia,M.E., Richelle,J. and Wodak,S.J. (1995) Optimal protein structure alignments by multiple linkage clustering: application to distantly related proteins. *Protein Eng.*, **8**, 647–662.

Branden,C. and Tooze,J. (1991) *Introduction to Protein Structure*. Garland Publishing, New York, London.

Brennan,R.G. and Matthews,B.W. (1989) The helix-turn-helix DNA binding motif. *J. Biol. Chem.*, **264**, 286–290.

Fischer,D., Bachar,O., Nussinov,R.and Wolfson,H. (1992) An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. *J. Biomol. Struct. Dynam.*, **9**, 769–789.

Gibrat,J.F., Madej,T. and Bryant,S.H. (1996) Suprising similarities in structure comparison. *Curr. Opin. Struct. Biol.*, **6**, 377–385.

Grimson,W.E.L. (1990) *Object Recognition by Computer—The Role of Geometric Constraints*. MIT Press, Cambridge, MA, USA.

Grimson,W.E.L. and Huttenlocher,D.P. (1990) On the sensitivity of geometric hashing. *Proceedings of the Third ICCV*, pp. 334–338.

Guéziec,A., Pennec,X. and Ayache,N. 1997. Medical image registration using geometric hashing. *IEEE Comput. Sci. Eng. Spec. Issue Geometric Hashing*, **4**, 29–41.

Harrison,S.C.and Aggarwal,A.K. (1990) DNA recognition by proteins with the Helix-Turn-Helix motif. *Annu. Rev. Biochem.*, **59**, 933–969.

Holm,L. and Sander,C. (1994) Searching protein structure datbases has come of age. *Proteins: Struct. Funct. Genet.*, **19**, 165–173.

Holm,L. and Sander,C. (1995) 3-D Lookup: fast protein structure database searches at 90% reliability. *Intell. Syst. Mol. Biol.*, **3**, 179–187.

Lamdan,Y. and Wolfson,H.J. (1988) Geometric hashing: a general and efficient model-based recognition scheme. *Proceedings of the Second ICCV*, pp. 238–289.

Lawson,C.L., Zhang,R.G., Schevitz,R.W., Otwinowski,Z., Joachimiak,A. and Siegler,P.B. (1988) Flexibility of the DNA-binding domains of TRP repressor. *Proteins: Struct. Funct. Genet.*, **3**, 18.

Lessel,U. and Schomburg,D. (1994) Similarities between protein 3-D structures. *Protein Eng.*, **7**, 1175–1187.

Madej,T., Gibrat,J.-F. and Bryant,S.H. (1995) Threading a database of protein cores. *Proteins: Struct. Funct. Genet.*, **23**, 356–369.

Mizuguchi,K. and Go,N. (1995) Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng.*, **8**, 353–362.

Mondragon,A., Wolberger,C. and Harrison,S.C. (1989) Structure of phage 434 Cro protein at 2.35 Angstroms resolution. *J. Mol. Biol.*, **205**, 179.

Orengo,C.A. and Taylor,W.R. (1996) SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol.*, **266**, 617–635.

Pennec,X. (1996) L'Incertitude dans les problèmes de reconnaissance et de recalage—applications en imagerie médicale et biologie moléculaire. PhD Thesis, Ecole Polytechnique, Palaiseau, France.

Pennec,X. and Ayache,N. (1994) An $O(n^2)$ algorithm for 3D substructure matching of proteins. In Califano,A., Rigoutsos,I. and Wolson,H.J. (eds), *Shape and Pattern Matching in Computational Biology—Proceedings of the First International Workshop.* Seattle, WA, June 20, 1994. Plenum Publishing, pp. 25–40 (also as INRIA Research Report no. 2274).

Pennec,X. and Thirion,J.P. (1997) A framework for uncertainty and validation of 3D registration methods based on points and frames. *Int. J. Comput. Vision*, **25**, 203–229.

Preparata,F.P. and Shamos,M.I. (1985) *Computational Geometry, an Introduction*. Springer Verlag, New York.

Raumann,E.E., Rould,M.A., Pabo,C.O. and Sauer,R.T. (1994) DNA recognition by beta-sheets in the ARC repressor-operator crystal structure. *Nature*, **367**, 754–757.

Remington,S.J. and Matthews,B.W. (1980) A systematic approach to the comparison of protein structures. *J. Mol. Biol.*, **140**, 77–99.

Rigoutsos,I. and Hummel,R. (1993) Distributed Bayesian object recognition. *Proceedings of an International Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 180–186.

Rossmann,M.G. and Argos,P. (1976) Exploring structural homology of proteins. *J. Mol. Biol.*, **105**, 75–95.

Sayle,R. and Bissel,A. (1992) RasMol: A program for fast realistic rendering of molecular structures with shadows. In *Proceedings of the 10th Eurographics UK'92 Conference*.

Thirion,J.-P. (1996) New feature points based on geometric invariants for 3D image registration. *Int. J. Comput. Vision*, **18**, 121–137.

Vriend,G. and Sander,C. (1991) Detection of common three-dimensional substructures in proteins. *Proteins: Struct. Funct. Genet.*, **11**, 52–58.

Wolfson,H.J. (1990) Model-based recognition by geometric hashing. In Faugeras,O. (ed.), *Proceedings of the 1st European Conference on Computer Vision (ECCV 90). Lecture Notes in Computer Science 427.* Springer Verlag, New York, pp. 526–536.